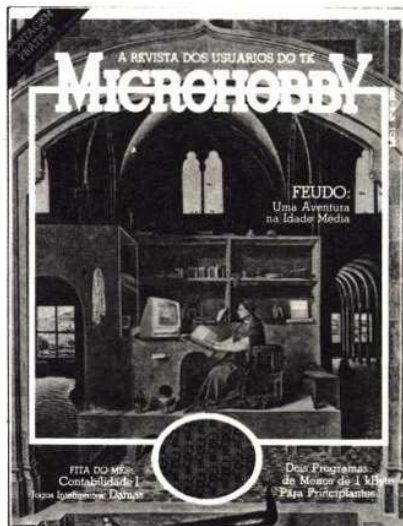
Expediente

DIRETOR EDITORIAL
Pierluigi Piazzi
JORNALISTA RESPONSÁVEL
Aristides Ribas de Andrade Fº
EDITOR
Álvaro A. L. Domingues
COORDENAÇÃO EDITORIAL
Ana Lúcia de Alcântara
ASSESSORIA TÉCNICA
Flavio Rossini
Wilson José Tucci
ANÁLISE E REDAÇÃO
Nancy Mitie Ariga
Carlos Eduardo Rocha Salvato
Renato da Silva Oliveira
Roberto Bertini Renzetti
ARTE
Cassiano Roda
Eliana Santos Queiroz
Fatima M. Rossini Gouveia
Osmère Sarkis
PRODUÇÃO GRÁFICA
José Carlos Sarkis
ILUSTRAÇÕES:
Rodival Matias
COLABORARAM NESTE NÚMERO
Eduardo J. V. Manso, Felipe Schmidt, Nilson D. Martello, Igor Sartori, Tanios Hamzo.
CORRESPONDENTES
Londres — Robert L. Lloyd
Paris — Alain Richard
New York — Natan Portnoy
Milão — Bruno Origo
PUBLICIDADE
Gina Elimelek
Atílio Debatin
Aurio J. Mosolino
Edson R. Silva
ASSINATURAS
Azarias Cordeiro dos Santos
MICROHOBBY é editada mensalmente pela **MICROMEGA** — Publicidade e Material Didático Ltda. — INPI 2992 Livro A. Endereço para correspondência: Rua Bahia, 1049 — Caixa Postal 60081 — CEP 05096 — São Paulo, SP. Telefone: (011) 256-8348. Para solicitar assinaturas (12 números), enviar cheque nominal cruzado à MICROMEGA PMD LTDA., no valor de Cr$ 14.800,00.
Tiragem desta edição: 30.000 exemplares

NÚMERO 6: (Dez. 83)

FOTOLITOS
Ponto Reproduções Gráficas SC Ltda.
Fone: 814-6311

IMPRESSÃO:
Gráfica Tania
Fone: 826-4011

Índice

CAPA
Hugo Faleiros e Cassiano Roda

Editorial

Na Feira de Informática, realizada no mês passado em São Paulo, foram vistas coisas interessantes, ridículas, opacas e brilhantes. Duas, porém, foram altamente preocupantes: a pressão que a IBM e a HP estão exercendo para entrar no mercado brasileiro de micros. Esta pressão esta sendo feita com a sutileza de uma revoada de hipopótamos, ou seja, esta aí para quem quer ou não vê-la. Isto nos faz refletir um pouco sobre a reserva de mercado na área de micros e sobre sua validade.

Existem fortíssimos argumentos contra esta reserva de mercado, mas por outro lado não faltam argumentos a favor, alguns dos quais até meramente emotivos mas não por isso menos válidos.

O próprio fato de ver o Roberto Campos (Bob Fields) lançar invectivas contra esta reserva de mercado já fez muitos brasileiros conscientes se posicionarem a favor da mesma.

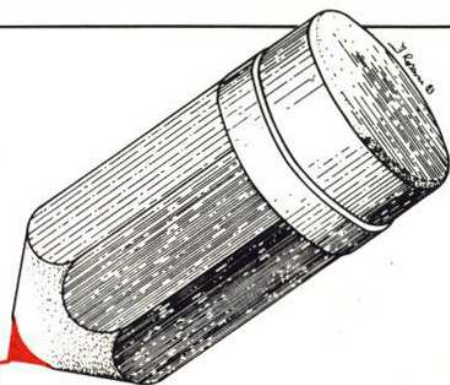
De qualquer forma é um assunto que merece muita ponderação e que não pode, de forma alguma, ser resolvido por uma irresponsável penada palaciana.

Qual deve ser a posição do usuário numa situação destas?

Ora, o que o leitor acharia de poder comprar um ótimo automóvel japonês pela metade do preço do mais barato carro brasileiro? Perspectiva tentadora não?

Agora o que este mesmo leitor acha que aconteceria com a já semi-falida economia brasileira se abríssemos a livre importação de carros estrangeiros?

Está portanto na hora de tomarmos consciência que a economia brasileira não é uma abstração: a economia brasileira somos nós! E se nós não cuidarmos do que é nosso, fatalmente efetivar-se-á de maneira definitiva o estado de semi-colonia em que já nos enfiaram.

CARTA DOS LEITORES

Pierluigi Piazzi e Equipe,

Venho por meio desta parabenizá-los pela conceituada revista **Microhobby,** que tem, já de início, feito por merecer um grande elogio de nossa parte — os TKmaníacos —, tanto pela beleza do material como pela preocupação em oferecer-nos o melhor material de cunho didático.

Continuem assim pois é disso que nós, TKmaníacos, estamos precisando: uma revista estruturada como a **Microhobby**.

Gostaria também de aproveitar a oportunidade para tirar um "grilo" no programa *Genius.* Neste programa, notei uma novidade (pelo menos para mim), a utilização de vários INKEY$ ao invés de INPUT, por exemplo:

```
1130 POKE 16437,255
1140 PAUSE 30000
1150 LET I$=INKEY$
1160 IF I$<"1" OR I$>"9" THEN GO
TO 1130
```

Agora pergunto: "Será que, se ao invés de LET I$ = INKEY$, fosse utilizado INPUT I$ na linha 115Ø, mudaria alguma coisa?" — "E nas linhas 149Ø, 173Ø, o que poderia acontecer ao programa?".

Fico aguardando uma resposta e gostaria também de trocar idéias com outros assinantes, através do envio de cartas, para mim, no seguinte endereço: *Marco Aurélio — Avenida Afonso Pena, 1557/214-B — Centro — Campo Grande — CEP 79100 — Campo Grande, MS.*

Caro Marco Aurélio,

Gratos pelos parabéns à **Microhobby.** *A melhor forma de responder sua dúvida é a seguinte:*

Substitua as linhas 115Ø, 149Ø e 173Ø por INPUT I$ e verifique por sí, qual o resultado.

Como achamos que você não deve gostar de nossa "melhor resposta", lhe adiantamos que ocorrerá mudança, muito pequena, pois enquanto INPUT aguarda a digitação de uma ou mais teclas seguida(s) de NEW LINE, INKEY$ aguarda a digitação de uma única tecla. Daí, em programas como Genius *preferir-se INKEY$ a INPUT (pelo menos por economia e rapidez).*

○

Prezados Senhores,

Possuo um TK82-C e venho tendo problemas com ele, pois, quando estou digitando um programa, após determinado tempo, começa a haver troca de caracteres de números de comando e, quando tento corrigí-los a imagem na tela desaparece sendo que, a única maneira de recuperá-la é desligando o micro (e perder o programa que estava sendo digitado).

Gostaria de saber se esse comportamento é comum, se for, que posso fazer para evitá-lo; se não, que posso fazer para corrigí-lo.

Derrel H. Santee — Brasília, DF.

Caro Derrel,

De modo algum esse comportamento de seu TK pode ser considerado "normal". Você deve encaminhá-lo à Microdigital para ser reparado.

○

Prezados Senhores,

Sou usuário do TK-82C a seis meses e assinante da revista *Microhobby,* a qual tenho acompanhado com crescente interesse pelos artigos e dicas que apresenta.

Tenho porém uma questão a propor que, no meu entender, deverá auxiliar diversos usuários e se relaciona com o controlador de jogos e seu interfaceamento com os TKs.

Ao comprar o controlador e interfaceá-lo com o TK-82C, notei que o botão de disparo não funcionava; ao pedir informações a um revendedor, este esclareceu que alguns modelos não estavam adaptados.

Gostaria que a revista publicasse alguma matéria sobre o assunto e se possível, como fazer a adaptação para aqueles que já estão fora do período de garantia.

Álvaro G.S. Barcellos — Icaraí — Niterói, RJ.

Caro Álvaro,

De fato, os primeiros micros TK-82 que surgiram no mercado não permitem o perfeito funcionamento do "Joystick". Para adaptá-lo, você deve procurar alguma oficina especializada ou enviá-lo à Microdigital. Quanto a publicação de matéria a respeito em nossa revista, terá a devida atenção e será submetida à apreciação de nosso editor, podendo ou não ser publicada.

○

Prezados Senhores,

Gostaria que me respondessem uma pergunta: "Me parece que, começando a gravação de um programa que tenha na primeira linha uma instrução DIM acontecem problemas. Será que é o meu TK ou isto acontece em todos os TKs?"

Eduardo Henrique Oliveira — São Paulo, SP.

Caro Eduardo,

A gravação de um programa do TK para fita deverá ocorrer normalmente, independentemente do que esteja na primeira linha do programa. Se com seu micro isso não ocorrer, sinal de que ele é o "diferente".

Junto com a carta, você nos enviou a resposta (correta) do Quebra-Cabeça Verdadeiro ou Falso *e perguntou-nos também, se a oitava asserção foi bem traduzida. Você tem nosso parabéns, pois a resposta é* sim *e essa nos pareceu a asserção de tradução menos evidente.*

○

Prezados Senhores,

Como assinante desta revista só posso elogiá-los. Está engrossando a cada número e mesmo que aumente minha maquinária, *jamais* deixarei de assiná-la, pois *jamais* venderei meu TK.

Mas, tenho uma dúvida: "que estória é essa de teclado musical para o TK? E ainda como posso fazer meu TK fazer "beep", "beep" (bip = sinal sonoro)?"

Outra coisa: poderei, um dia, adaptá-lo para telefone, a fim de saber minha conta bancária, etc?

Sou médico-anestesista e gostaria de trocar (comprar, também) programas na área. Escrevam, colegas. *Dr. Roberto Araujo — R. Uruguay, 205, apto. 803 — CEP 20510 — Rio de Janeiro, RJ.*

Caro Dr. Roberto,

Antes de mais nada, obrigado pelos elogios.

O "teclado musical para o TK" nada mais é que um programa que faz com que algumas teclas, quando digitadas, produzam sons semelhantes a notas musicais. Quanto aos "beep's" e as "interfaces" para telefone, podem ser conseguidos através de periféricos do tipo das TKPLAK's (vide nº 3 de **Microhobby***).*

Prezados Senhores,

Tenho acompanhado a **Microhobby** desde o seu lançamento, e gostaria de parabenizá-los pela maneira objetiva e didática utilizada em todas as matérias, que permite aos leigos como eu, um acompanhamento integral dos assuntos abordados.

Eu tenho algumas dúvidas que talvez vocês pudessem esclarecer. São elas:

1. Os periféricos existentes no mercado inglês e outros para os ZX's (tipo *Memotech*, etc.) são, eletronicamente, totalmente compatíveis com os TK's?

2. Na edição nº 3 de **Microhobby**, na seção "Novidades" vocês apresentaram o TPLAK, mas não o nome do fabricante, como, quando e por quanto se pode adquirí-lo. Gostaria de ter maiores informações a respeito.

3. Ao converter para fita alguns programas editados em revistas nacionais e estrangeiras, costumo fazer incrementações nos mesmos, para torná-los mais interessantes — geralmente nos programas "Aplicativos". Minha pergunta é a seguinte: Vocês aceitariam como colaboração estes "Programas Modificados", embora, a base seja de um outro programador (que nem eu mesmo conheço) e editado em outra revista?

4. Gostaria de saber se é do conhecimento de vocês o lançamento do microcomputador a cores, acessível em preço como foi o TK, por parte da Microdigital?

Desde já, grato por este papo e por suas possíveis respostas.

Armando C.F. Pires dos Reis — Rio de Janeiro, RJ.

Caro Armando,

Suas dúvidas são bem simples e acreditamos que a 2ª e a 4ª já foram respondidas na edição nº 5 de **Microhobby**. *Abaixo respondemos as demais:*

1ª — Todos os periféricos para os ZX's que conhecemos são eletronicamente compatíveis com o TK. De qualquer modo, é bom que antes de adquirí-los você procure conhecê-los bem.

3ª — Temos como certo que nada se cria a partir do nada. Obviamente se os programas "originais" forem incrementados ou modificados o suficiente para serem diferentes dos originais, eles não serão mais simples cópias, mas sim outros programas "originais". Nesse caso, nós os aceitaremos com muito prazer, desde que você nos indique a origem do programa "original", ou seja, o nome da publicação de onde ele foi extraído e também do autor, se houver.

Prezados Senhores,

Estou tentando ser autodidata em meu TK-82 (sem expansão, sem SLOW) nas horas vagas. Foi com satisfação que recebi a fita da **Microhobby**, porém não consigo rodar o programa da *Pulga*.

1. Este problema ocorre em função do SLOW?

2. Será que poderiam enviar-me a listagem correta para que possa checar?

3. Algumas instruções me são muito estranhas: por exemplo na linha 25 LET A$ = "█" + A$ (TO31), que tipo de definição é essa? O manual da máquina não me auxiliou muito.

4. Quero me aprofundar (iniciar) com PEEK, POKE e USR. Já há alguma publicação da Micromega à disposição?

Francisco José Pall — São Paulo, SP.

Caro Francisco,

É muito bom você tentar aprender sozinho a "conversar" com seu TK. Essa é, em nossa opinião, a melhor forma que há para isso. Claro que, quando surgem dúvidas às quais não se tem tempo ou paciência para esclarecê-las, é conveniente procurar auxílio.

Respondemos, agora, sua primeira, segunda e quarta dúvidas. A terceira será respondida na seção **Desgrilando**.

1) Certamente, sim. Para amenizar o problema, retire a linha:

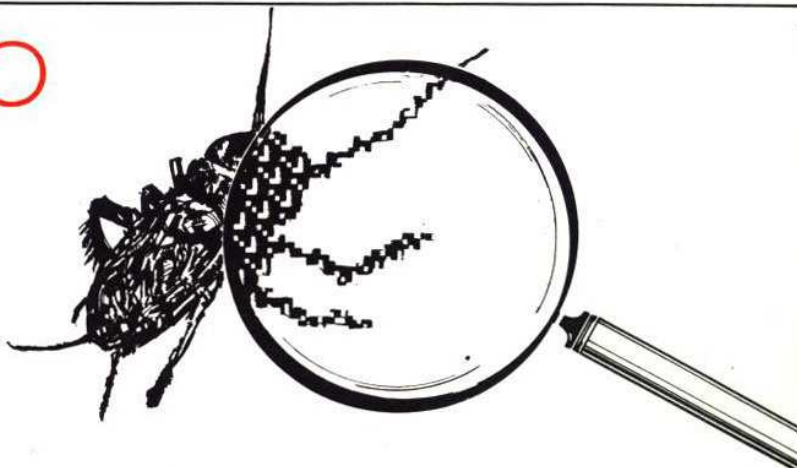
```
2 SLOW
```

e introduza:

```
36 PAUSE 20
51 PAUSE 20
76 PAUSE 20
```

2) Com estas modificações, seu programa rodará perfeitamente. A listagem torna-se desnecessária, uma vez que o programa que enviamos na sua fita-brinde está correto.

4) O livro de Flavio Rossini, **Linguagem de Máquina para o TK**, *é o que indicamos para o seu caso, bem como de outros leitores que desejam conhecer este novo campo de aplicações para o seu computador.*

DESGRILANDO

Prezados Senhores,

Parabenizo vocês por esta ótima revista, que está me ensinando o Z80 e aumentando o meu conhecimento sobre o BASIC.

Acho muito interessante a seção *Dicas* e gostaria de mais uma: "Como é que coloca a linha zero" que, em certos casos, tem o nome do autor do programa e nunca pode ser retirada. Grato pela atenção.

Edson Moreira Menezes – Aracaju, SE

Caro Edson,

Obrigado pelos parabéns.

Você nos perguntou como se introduz uma "linha 0", isso é muito simples. Digite a linha:

```
1 REM "MICROHOBBY"
```

Agora digite (sem número de linha):

```
POKE 16510,0
                (E 2 VEZES NEW LINE)
```

Aí está a sua linha 0!

Se quiser mudá-la para `1 REM "MICROHOBBY"` *novamente, basta fazer* `POKE 16510,1 (E NEW LINE)`

Ao contrário do que muitos pensam, a principal utilidade de uma linha `0 REM ...` *não é "gravar" nomes, marcas ou títulos, mas sim subrotinas em linguagem de máquina. Muitas vezes essas linhas ao serem trazidas para a parte inferior da tela com EDIT, se auto-destroem parcialmente (veja a seção* **Desgrilando** *de* **Microhobby** *nº 5). Colocando-as na linha* `0 REM ...` *evita-se editá-las acidentalmente.*

Se você está interessado em colocar uma "marca" sua num programa, existem métodos bem mais eficientes. Como por exemplo, digite:

```
  1 PRINT AT 20,0;"                MI
CROHOBBY               ";AT 21,0;"
        SEU NOME                  "
```

Após isso, digite direto:

```
POKE 16509,63        (E NEW LINE)
```

e um programa qualquer, por exemplo:

```
1 FOR N=0 TO 255
2 PRINT CHR$ N;
3 NEXT N
```

A sua listagem agora deve ser:

```
   1 FOR N=0 TO 255
   2 PRINT CHR$ N;
   3 NEXT N
G129 PRINT AT 20,0;"                MI
CROHOBBY               ";AT 21,0;"
        SEU NOME                  "
```

Rode-o e veja o que ocorre.

Tente apagar a linha que "imprime" sua marca sem apagar o programa.

Provavelmente você não conseguirá!

Pode-se, ainda, substituir o número 63 em

```
POKE 16509,63
```

por qualquer outro número entre 40 e 64 (inclusos).

Trecho da carta de Francisco José Pall, publicada na seção **Carta dos Leitores** *deste número:*

"Algumas instruções me são muito estranhas: por exemplo na linha 25

```
LET A$="█"+A$( TO 31)
```

que tipo de definição é essa? O manual da máquina não me auxiliou muito.

Caro Francisco,

Na linha

```
LET A$="█"+A$( TO 31)
```

está se atualizando o "string" A$. A partir dessa linha, ela terá como primeiro caractere o espaço reverso, e como caracteres subsequentes, todos os outros que ela já possuia, até o 31º.

Para que isso fique claro, digite as linhas a seguir e "rode-as":

```
  10 LET A$="ABCDEFGHIJ"
  20 PRINT "A$=";A$
  30 PRINT AT 2,0;"A$(1 TO 5)=";
A$(1 TO 5)
  40 PRINT AT 4,0;"A$( TO 5)=";A
$( TO 5)
  50 PRINT AT 6,0;"A$(6 TO 10)="
;A$(6 TO 10)
  60 PRINT AT 8,0;"A$(6 TO )=";A
$(6 TO )
  70 PRINT AT 10,0;"A$(4 TO 7)="
;A$(4 TO 7)
```

Como você pode notar pela imagem na tela (figura) a expressão A$ (1 TO 5) é a mesma que A$ (TO5) e A$ (6 TO10) é a mesma que A$ (6TO). Experimente mudar os números dentro dos parênteses e verifique os vários casos possíveis.

Cada expressão A$ (m TO n) — onde m e n são números naturais — é uma **sub-string** ou **slicing** da **string** A$. Nessa **slicing**, o primeiro caractere é o m-ésimo elemento da **string** A$ e o último caractere é o n-ésimo elemento da **string** A$.

Na linha

```
LET A$="█"+A$(1 TO 31)
```

está se fazendo uma "concatenação" (junção ou união) de duas **strings**. A primeira é constituída apenas pelo caractere " █ " e a segunda é uma **slicing** que possui do 1º ao 31º elemento de A$.

Acrescente ao programinha anterior as linhas adiante:

```
 80█LET A$="█"+A$( TO 9)
 90 PAUSE 180
100 GOTO 20
```

Rode-o e veja o que acontece. Quando o micro lê pela primeira vez a linha 80 ele atualiza a **string**

```
A$="ABCDEFGHIJ"
```

para

```
A$="█ABCDEFGHI"
```

Da linha 100 ele retorna à linha 20 e refaz a tela. Ao passar pela segunda vez pela linha 80, atualiza novamente A$.

Assim, após ler 10 vezes a linha 80, teremos:

```
A$="██████████"
```

Para finalizar, experimente rodar o programinha mostrado a seguir.

Ele faz sua **string** "girar".

Procure entender como.

```
10 LET A$="█ █ █ █ █ █ █ █ █ █"
20 PRINT AT 12,6;A$
30 LET A$=A$(18)+A$( TO 17)
40 GOTO 20
```

ERROS DE IMPRESSÃO DO LIVRO LINGUAGEM DE MÁQUINA PARA O TK VOLUME I

Flavio Rossini

Infelizmente, no livro "Linguagem de Máquina para o TK", alguns erros de impressão passaram despercebidos, o que torna aparentemente sem sentido algumas passagens. Devido à impossibilidade prática de se adicionar uma "errata" ao livro, vou publicar aqui na Microhobby os erros, contando com a colaboração de vocês para que os divulguem para quem o possui. Obviamente não mencionarei erros "supérfluos" como, por exemplo, a troca de uma letra minúscula por maiúscula.

Na página 70 temos o programa 4.8 que, aparentemente, não tem lógica. Basta modificar as seguintes linhas:

```
3065 IF INKEY$="" THEN GOTO 3065
3075 IF B=23 THEN LET B=1
3085 IF C=0 THEN LET C=22
3095 IF B$="6" THEN PRINT AT 0,U
SR 30000;A$(C)
4000 IF B$="7" THEN PRINT AT 21,
USR 31000;A$(B)
4015 GOTO 3065
```

Na página 91 na figura 5.6 os bytes na memória são, respectivamente, '39' (e *não* '94') e '75'; de fato, o CALL coloca no topo da pilha o endereço de retorno da subrotina, ou seja, 30009 (= '7539') que corresponde à primeira instrução INC HL.

Na página 127, no programa 8.2, falta uma aspa (") na linha 120:

```
120 PRINT "E AGORA VOU ME AUTOD
ESTRUIR","ADEUS...,SNIFF...SNIFF.
.."
```

Na página 148, no programa 9.2 a terceira linha está deslocada para a direita! De fato, LOOP é um *label*.

Na página 149 a linha 2075 deve ser substituída:

```
2075 GOTO 2050
```

Na página 159, na 10ª linha a palavra *absoluto* deve ser substituída por *aleatório*.

Na página 172, o final da 3ª linha deve ser substituída por "ao programa 10.4".

Finalmente no capítulo 11 todas as referências a figuras nas páginas 184 e 185 devem ser acrescidas de 3 unidades; por exemplo: figura 11.3 → 11.6. (Na figura 11.6 o número 4300 deve ser retirado).

Peço a compreensão de todos vocês e procurarei me esforçar ao máximo para que o segundo volume do livro saia o mais perfeito possível, exigindo uma revisão mais rigorosa por parte da Editora Moderna. Ele deverá ser lançado nos primeiros meses de 1984 e terá como objetivos principais:

– finalizar as explicações quanto às instruções do Z80 (p. ex.: DAA, RLD, IN, OUT, EI, etc.)
– explicar e apresentar os programas ASSEMBLER e DISASSEMBLER (&MONITOR), e utilizar os mesmos para colocar programas complexos (por exemplo, jogos) no TK e estudá-los. Portanto, utilizarei os programas produzidos pela MULTISOFT que já se encontram no mercado.
– explicar detalhadamente a estrutura da ROM, abordando todos os comandos em BASIC e inclusive as rotinas de floating point, para podermos utilizar as subrotinas da ROM nos nossos programas.
– dar uma noção de hardware, explicando suscintamente como é a arquitetura interna do TK, através do estudo do funcionamento do Z80.

Espero que meu esforço em escrever esses livros sirva para suprir a curiosidade e vontade de aprender de todos vocês que, como eu, "curtem" muito o TK.

De qualquer modo, coloco-me à disposição para esclarecer, por carta, qualquer dúvida eventual que vocês possam ter. Basta escrever para a MICROMEGA numa carta endereçada ao "Departamento de Assessoria Técnica – at. prof. Flavio Rossini". Procurarei responder todas as cartas no menor tempo possível e, caso houver muitas dúvidas "comuns", o esclarecimento será publicado na revista na seção "Desgrilando".

Mais uma vez, obrigado pela atenção.

INFORMÁTICA 83:

Durante uma semana consecutiva o espaço formado pelo Parque Anhembi e o Palácio das Convenções foi frequentado por uma quantidade de pessoas estimadas em aproximadamente 300 mil. O público presente ao Congresso e a Feira, superaram os números calculados por seus organizadores. Vieram crianças com seus pais, estudantes, especialistas da área, intelectuais, diversas autoridades políticas de dentro e fora do país -- todos interessados com o futuro da informática no Brasil.

Estiveram presentes na feira, cerca de 319 expositores num total de 260 estandes distribuídos nos 22 mil m² do Anhembi. O Congresso recebeu 5 mil congressistas, foram apresentados 120 trabalhos técnicos – 25 do exterior – teve seis mil pessoas inscritas sendo 2.400 estudantes (200 da América Latina).

Este evento por sua dimensão internacional e nível de frequência diária tornou-se o acontecimento de maior destaque, senão nacional, pelo menos a nível estadual.

Além do Congresso e da Feira, montou-se um *circo informático* ao lado do Palácio das Convenções com o objetivo de trazer as crianças na faixa de 12 a 16 anos, para conhecerem um computador. No meio do circo, estava o picadeiro com o palhaço ***Abracadabra*** animando cerca de 400 crianças trazidas de oito escolas diferentes, juntamente com os computadores M-LOGO cedidos pela Unitron. As crianças maravilhadas, divertiram-se durante os cinco dias de funcionamento do circo, traçando figuras geométricas e funções trigonométricas na tela dos computadores.

O tema predominante em todas as palestras no Congresso e nas conversas entre empresários e visitantes de diversos estados e países, na feira, giraram em torno da tão afamada política de ***reserva de mercado*** traçada pelo governo federal.

Logo de início, na abertura do Info/83 realizada em sessão solene no Palácio das Convenções, presidida pelo ministro Danilo Venturini e contando com a presença de diversas autoridades, entre elas o secretário da Indústria, Ciência e Tecnologia do Estado de São Paulo, Einar Kok, o ponto central dos discursos foi a necessidade do País obter capacitação tecnológica e domínio das técnicas empregadas.

A pressão exercida por multinacionais presentes na Feira, foi outro ponto que demonstrou a necessidade da preservação de mercado. Vontade é que não falta, por parte de multis, em penetrarem neste mercado, como por exemplo a IBM que este ano trouxe à feira, todo seu potencial tecnológico, baseado numa tremenda campanha publicitária em torno da chamada "Tecnologia, Aqui, Agora e para o Futuro".

1. O estande da Micromega e o famoso gorila

Em contrapartida o Info/83 serviu, também, para chamar a atenção para outros pontos que merecem ser ressaltados, como, por exemplo, procura de uma forma de associação entre empresas nacionais e internacionais, através, por exemplo, de "Joint-Venturi" –, uma proposta dada pelo ministro Camilo Penna.

Desta vez, o objetivo dos organizadores do evento foi alcançado, afinal, a frequência, ao menos na ***feira,*** superou as expectativas e desta vez, a popularização do computador foi algo concretizado. Esta popularização irá ampliar o mercado, gerando sua própria transformação.

Outro fator demonstrado, é que o povo – ao menos desta vez –, apoia uma das medidas econômicas adotadas pelo governo federal. Em todos os dias percebeu-se que, todos os setores da sociedade, através de manifestações em jornais e depoimentos emitidos, estão totalmente favoráveis à política de reserva de mercado.

A Feira

A feira de Informática deste ano poderia ter como chamada o seguinte: "Venha à Disneyworld da Informática Brasileira!". Foi um verdadeiro festival de atrações múltiplas.

a Feira e o Congresso

Ana Lúcia de Alcântara

Logo na entrada o visitante se defrontava com alguns computadores cedidos pela Itaútec colhendo informações com o objetivo de traçar, em seguida, o ***perfil do visitante.*** Uma coisa difícil de ser conseguida, pois o público ali presente foi bem variado.

A feira esteve distribuída em setores: microcomputadores, microprocessadores, videogames, suprimentos e periféricos, software, serviços, telecomunicações, comércio, copiadoras, máquinas de escritórios, instrumentos e mobiliários e universidades.

Penetrando no enorme pavilhão do Anhembi, o visitante via, de imediato, os estandes da Prodam, fornecendo informações sobre turismo e lazer e o de artes computacionais expondo obras de artes feitas em computador. Neste estande estavam algumas empresas fornecedoras de serviços de video texto, como a Editora Nobel com seu serviço "Arte on Line" e a Intergraph Sistemas apresentando obras de vários artistas.

Apreciando as maravilhas apresentadas, o visitante assustava-se, de cara, com a enorme quantidade de pessoas que passeavam ou brincavam nos computadores espalhados nos diversos estandes. Andando pelas ruas de A a Z este mesmo visitante ia recebendo bexigas, pipocas, cataventos, pastinhas, sacolinhas, broches, folhetos explicativos, adesivos, papeizinhos para concorrer a sorteios de micros, além de ir vislumbrando moças bonitas sorrindo graciosamente sempre presentes nos estandes.

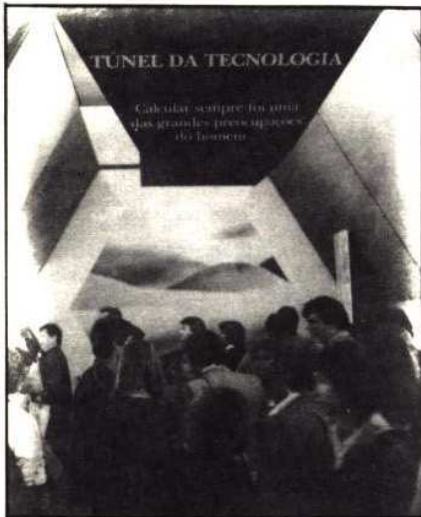
Conforme a distribuição da feira os grandes estandes ficaram concentrados no final. Lá estavam a Cobra, IBM, Xerox, Sisco, Scopus, Prológica, SID-Sharp, Elebra, Labo, Dismac, Itaútec, antecedidos de outros menores, mas que marcaram sua presença tanto em termos de frequência de público, como em apresentação de novidades. Um destes estandes foi o da Microdigital fixada nas proximidades dos estandes da Polymax, Sysdata, Ritas e Unitron entre outros que, com seu estande vermelho e preto, marcou presença durante toda a feira. Distribuindo micros e coca-cola todas as noites, além de recepcionistas vestidas de prata que colavam adesivos em todos os que passavam pelo estande, a Microdigital expôs seus TKs, com força, lançou seu videogame — o "Onyx" — e apresentou ao público, dois protótipos do TK-2000 color.

A Xerox distribuia cópias ampliadas de Charles Chaplin enquanto a Nashua cópias do Cebolinha e da Mônica.

Quem chegasse ao final da feira se defrontava com dois chamativos estandes. O primeiro por sua suntuosidade — o da IBM e o segundo — o da SISCO, por sua iluminação e um detalhe bem chamativo: o som de um foguete decolando. A SISCO trouxe ao Anhembi seu estande iluminado com fios de ***neon*** montado com a mesma concepção que o estande da IBM: "túnel do tempo". Uma gravação com o som do foguete Saturne decolando era acionada de 15 em 15 minutos juntamente com uma voz computadorizada que dizia: SISCO! SISCO!. . . Este som fêz os ouvidos ao lado tremerem, forçando a SISCO diminuir o som, e a expandir o tempo de acionamento da "simpática" voz.

Além do som interessante, a SISCO trouxe 12 lançamentos na área de micros para apresentar na feira, entre estes, o MS-800 ou "grilo" com 128 kByte e memória RAM.

O vizinho da SISCO — a IBM — foi o centro das atenções de toda a feira. Seu estande — o maior —, com 960 m² mais os 200 metros adicionais onde ficaram os CPU's, foi criado em cima do logotipo "Tecnologia, Aqui, Agora e para o Futuro". Lá, foram distribuídos 100 mil sacos de pipocas, 20 mil balões e 20 mil cataventos além dos interessantes e concorridos broches onde se via a figura de Charles Chaplin "mexendo" num computador.

Montado estrategicamente no final da feira no percurso onde todo visitante era →

2. O túnel no estande da IBM

obrigado a passar, a IBM trouxe dos EUA especialmente para a feira, máquinas que não podem ser comercializadas no Brasil — graças a reserva de mercado — como um terminal de painel a gás, o robô industrial 4281 e 12 micros PCs, que estavam abertos para demonstração.

Em contrapartida à pressão da IBM, lá estava o estande preto da Prológica montado em cima do logotipo: "Reserva de Mercado — Defesa de valores nacionais". Também distribuindo brochinhos onde se via o logotipo de sua campanha publicitária, a Prológica trouxe à feira a versão modular de seu microcomputador Sistema-700, as impressoras P-740 e Qualitat além dos CP's com som estéreo e joy control, programados com jogos eletrônicos expostos para usufruto dos sediosos e curiosos visitantes.

A SID (dando destaque ao multiprocessor — chamado Sistema 8 e o concentrador 3510) foi um dos estandes, juntamente com: a SCOPUS que trouxe o Nexus 1600, a Softec com o "Ego" e a Itaútec com o micro "Júnior", que estiveram em termos de tamanho e frequência de público, no mesmo porte que o da Prológica.

A grande vedete além dos grandes estandes, foram realmente os microcomputadores. As grandes novidades estiveram nesta área além, é lógico, dos videogames, que tiveram a maior receptividade por parte da garotada.

O TK-83 da Microdigital, o TK-2000 color, o micro 8221 da Labo, o Fox da Brascom, o Ap II para monitor colorido da ***Unitron,*** o Poly 150 WP um micro voltado ao processamento de textos, da Polymax, o Alpha 2064 MH destinado a trabalhos em redes e o D8100 com saída para TV a cores e preto e branco, da Dismac, foram alguns lançamentos na área de microcomputadores.

Os estandes que lançavam videogames tiveram, durante toda a feira, uma frequência inaudita e esperada. Afinal, jogos e, ainda mais coloridos e barulhentos, atraem até mesmo o menos curioso visitante.

A Milmar fabricante de Apple II Plus lançou o Dactari, a Microdigital o Onyx, a Splice, o Splice Vision e o famoso Atari esteve distribuído em diversos estandes na feira, causando bastante rebuliço entre a criançada sediosa por jogar nas terríveis maquininhas.

Outra atração na III Feira de Informática foram as editoras. Várias revistas foram lançadas. A Editele lançou a Bits, a Prológica a "Geração Prológica" e a Micromega a "Pinball".

Uma curiosidade atiçou os visitantes e expositores até o quarto dia da feira: O que um gorila fazia numa feira de informática?!... O gorila, uma versão brasileira do King-Kong esteve de segunda a domingo em cima do estande da Micromega e gerou burburinho nas conversas de muita gente. Na quinta-feira, o segredo foi desfeito quando o gorila surgiu com uma faixa pendurada em seu corpo escrita: "PINBALL". Descobriram neste momento, que se tratava do lançamento da nova revista da Micromega que, fugindo um pouco dos micros lançará a Pinball -- uma revista de videogames.

Das inúmeras atrações da "feira-alegria" fatos importantes devem ser ressaltados como por exemplo a presença estrangeira com representações da Argentina, Chile, Colombia, Equador, Panamá, Venezuela e EUA, expondo na feira, a formação da empresa Informática Latino-Americana S/A formada com o intuito de atingir o mercado latino-americano.

O Congresso

O 26º Congresso Nacional de Informática foi marcado, essencialmente pelo teor político que adquiriu. Contando com a presença de autoridades de forte peso político desde secretários estaduais até o presidente da República, recebeu a visita de diversos congressistas e autoridades de vários países que proferiram palestras e discursos quase sempre voltados para a necessidade de preservação de um mercado latino-americano.

Além das diversas palestras (65 de expositores, 100 técnicas, sendo 12 internacionais), houve também diversos minicursos com o objetivo de proporcionar a familiarização do leigo com os micros.

A presença do presidente Figueiredo reafirmou a posição da maioria, quando ele afirmou: "ser imperativo abrir espaços para resultados mais concretos na formação de um modelo brasileiro de informática através da fabricação e equipamentos nacionais".

Uma das coisas que o Congresso mostrou este ano, foi o interesse de pessoas leigas pela área de informática pois dos cinco mil participantes, três mil não tinham contato com a tecnologia da área.

O objetivo de seus organizadores como na feira, foi totalmente atingido. Primeiramente pelo teor político que alcançou — todas as tendências estiveram presentes ao Congresso ou se preocuparam em seguir o curso que tomava as palestras nos jornais. Num segundo momento, mostrou-se, principalmente à área federal, a importância da regulamentação da profissão e o sindicalismo. ○

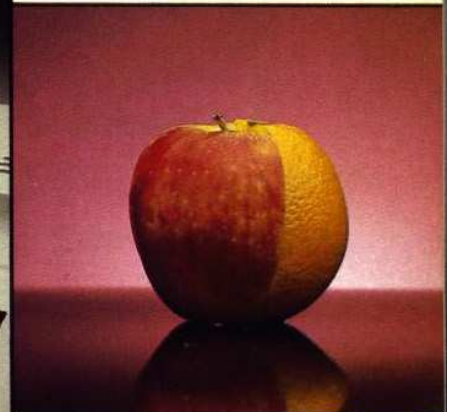
3. As crianças nos computadores do Circo Informático

I love apple.

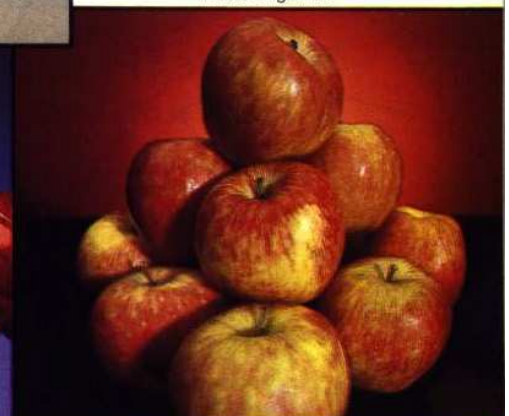
A SPECTRUM acertou em cheio o alvo quando resolveu mirar o Micro Engenho no Apple II. É o sistema mais utilizado por empresários, executivos e profissionais do mundo todo.

O Micro Engenho está entre os melhores computadores do mercado. Mas só ele oferece duas vantagens de dar água na boca: é fabricado com tecnologia Scopus, e executa todo e qualquer programa existente para o Apple II Plus.
Está explicado por que o Micro Engenho é uma escolha natural.

O Micro Engenho seguiu direitinho a receita do Apple II Plus, com um ingrediente a mais: é o único capaz de reproduzir todas as cores do Apple II nos televisores nacionais (PAL-M), dispensando quaisquer adaptações, tanto no micro como no TV.

Usar o Micro Engenho é tão fácil como fazer um milk shake.
Você coloca gráficos, cálculos, textos e ele entrega tudo bem mastigadinho.
Nada mais gostoso.

Além de ser inteiramente compatível com o Apple II Plus, o Micro Engenho oferece mais uma opção para quem também gosta de laranja: o módulo CP/M.
Experimente. Você vai se apaixonar pelo Micro Engenho.

O Apple II é um computador mundial mais conhecido que sorvete. E a mesma qualidade que o faz tão popular está no Micro Engenho. Prove. Você vai ficar derretido por ele.

A SPECTRUM põe o maior carinho em cada Micro Engenho que produz. Este amor continua durante o período de garantia e permanece sempre com o suporte de assistência técnica.

O Micro Engenho cresce com você.
A SPECTRUM oferece toda uma safra de opções e acessórios que atendem às mais diferentes necessidades, proporcionando uma ótima colheita de interfaces, moduladores, acionadores de jogos, aplicativos etc.

SPECTRUM R. Félix Guilhem, 913 - Tels.: (011) 260-0826 e 260-2551 - CEP 05069 - São Paulo - SP.

Colucci

*Na número 3 de **Microhobby** mostramos na seção **Programas do Mês** como implementar um arquivo no TK. Neste número recebemos uma colaboração extremamente interessante e divertida, que complementa aquele artigo publicado na revista 3 e mostra algumas sugestões e uma aplicação bastante apetitosa!*

APLICAÇÕES MAIS DINÂMICAS PARA O ARQUIVO

Nilson D. Martello

Um mesmo programa trabalhado por duas pessoas, mostrará algumas soluções comuns. No geral, entretanto, cada programador optará por soluções diferentes, com maior ou menor eficiência e, até mesmo, maior ou menor elegância.

Por outro lado, as tendências para programas mais econômicos (afinal, 16 kBytes não é quase nada!) levam a exercícios ginásticos na economia de bytes que, por vezes tornam-se ridículos, como no exemplo que segue:

"**1ØØ LET N = 1** – substitua por:
1ØØ LET N = PI/PI e veja quantos bytes foram economizados." Só que, na instrução 11Ø lê-se:

11Ø PRINT "EU NÃO VEJO COMO SERÁ LEVADO ATÉ LÁ DE MANEIRA SIMPLES ;

O que foi economizado na linha 1ØØ o autor jogou fora na instrução seguinte, num verdadeiro discurso.

Por fim, o ***emprego*** que se dá a um programa é característico da filosofia de cada um. Assim, discordo da utilização do microcomputador como simples agenda ou livro horário; prefiro uma agenda de papel e uma boa caneta esferográfica. É simples, rápido, direto e econômico.

Todavia, sendo um glutão nato, adorando bons pratos e acepipes; frequentando restaurantes sempre que possível, eu e minha esposa, por vezes, nos defrontamos com problemas que - estes sim! – são lindamente resolvidos pelos micros.

"Onde vamos comer hoje à noite?" – pergunta um de nós.

"O que V. deseja comer?" – responde o outro.

"Não sei . . . O que V. diz?" – volta o primeiro, e ficamos nesse chove-não-molha, vai-não vai, – a solução difícil de ser encontrada.

Hoje tudo é mais fácil com o emprego do programa ARQUIVO, publicado em Microhobby nº 3. Basta fazermos a classificação dos restaurantes já conhecidos, listá-los e escolher.

– Espere aí! – reclama o leitor. – Você não disse que preferia uma agenda de papel e uma caneta esferográfica?!

Exatamente, aí está o fulcro da questão: a vantagem do micro é ***selecionar*** para nós, muito além de uma simples listagem. Veja como classifiquei meu arquivo RESTAURANTES.

Quando o micro indaga "Deseja quantos campos?" escolhi quatro. Cada um deles com um máximo de quatro caracteres. De onde surgem os títulos:

NOME
TIPO
CR$
R.

→

Em NOME classificamos — obviamente — os restaurantes pelo nome mais significativo. Se o proprietário do restaurante tiver dotes verborrágicos, problema dele: "TRATTORIA LA BELLA ITALIA DI MAMMA ESGUIGNOTA" ficará reduzido a "Bella Italia".

Em TIPO empregaremos um campo de 6 caracteres: CARNE, PEIXE, MASSAS, VÁRIOS, ÁRABE, etc.

Em Cr$ queremos saber a faixa de preço (com a inflação não adiantará detalhar). Então partimos para o ALTO, MÉDIO, BAIXO, COM. (comercial). Neste último entram as casas que servem um menu fixo por dia, não necessariamente barato.

Por fim, em *R.* — por rua, avenida, etc. — o endereço. Este é um dos que ocupa espaço 'prá danar', não sendo raros os nomes "Peixoto Gomide de Oliveira Campos, nº 1934". Neste item, parti da premissa de que ***já conhecemos*** o restaurante — ou não estaria em nosso arquivo — bastando um ***lembrete*** do endereço: "R. PEIX. GOMIDE, 1934".

Então perceba, agora, a vantagem de um programa como ARQUIVO! Uma vez construído o RESTAURANTES, ao se ter dúvidas quanto a um programa, pedimos uma busca de registro. O micro nos pergunta: "Busca por qual campo?". Se a carteira está mais vazia, escolhemos o ítem CR$. Se as pessoas "não sabem" ou não estão "inspiradas" quanto ao que se deseja comer, buscamos por TIPO.

Como último comentário — e ligado ao início deste artigo discordamos da forma como foram tratadas as frases no programa ARQUIVO, seja no original de Russel King ou na versão paulista. O TK-82 (e seus primos) é muito lento e leva um tempo ***enorme*** escrevendo na tela. Ao mesmo tempo, cada letra é um byte e vale economizar memória para a aplicação que desejamos. Desta forma, sem prejuízo da inteligibilidade, restringimos o que vai nos PRINT do programa original.

Em lugar de REGISTROS, colocamos DADOS que significam a mesma coisa, economizam 4 bytes à cada vez que surgir, e é escrito de forma mais rápida. NUMERO, é claro, foi reduzido para Nº (no TK empregando o mesmo sinal gráfico SHIFT-1). De outro lado, nos pareceu desnecessário o ítem 6 (ordenar registros) e 7 — gravá-los. Consta da apresentação do menu:

6. NÃO TECLE RUN
7. P/ GRAVAR TECLE GOTO 999Ø.

Note outra economia: ***tecle,*** ao invés de ***digite.*** Não entrarei em outros detalhes, porém ordens como "DESEJA QUANTOS CAMPOS?" foi reduzida para "QTOS CAMPOS?", enquanto "QUANTOS CARACTERES PARA O TITULO DE CADA CAMPO?" (ufa!) para uns míseros e rápidos:" QTOS CARACTERES P/ CAMPO?".

Resultado final: tenha mais que uma simples agenda. Tenha um micro auxiliando-o no cotidiano. Por exemplo, hoje à noite iremos comemorar a aceitação deste artigo na . . . deixe-me ver. . .

"BUSCA POR TIPO?"

Eu teclo um "S".

MASSAS . . .
CARNES . . .
PEIXE . . ."

Consulto a esposa; ela está a fins de uma lagosta na manteiga com um vinho branco alemão.

Pego o telefone e consulte nosso editor, o Pierluigi . . .

— Você está doido? — estoura ele, do outro lado da linha. — A MICROHOBBY ainda não está pagando em dólares!

Desligo o telefone, tristonho, e volto ao microcomputador. Desta vez escolho o campo . . .

"CR$ . . . "

JOGOS INTELIGENTES

Felipe Schmidt

Em que consiste Inteligência Artificial? A principal meta da Inteligência Artificial (IA) é tornar os computadores *espertos.* Pesquisadores em IA escrevem programas e tentam fazer tarefas que ordinariamente poderiam ser realizadas unicamente pela razão e raciocínio humano. Embora ninguém tenha sido ainda levado a descobrir um programa que transforme o computador em uma entidade inteligente, algumas conquistas da IA são extraordinárias e excitam nossa mente com suas possibilidades.

O termo *Inteligência Artificial* abrange um considerável número de tópicos. Vejamos alguns deles:

• *Processamento da linguagem natural:* Tenta fazer com que os computadores entendam a linguagem humana e suas várias línguas, seja Inglês, Russo ou Português. Tem aplicação em, por exemplo, tradução por computador.
• *Solução de problemas:* É um campo importante. Podemos facilmente escrever um programa que resolva equações polinomiais, mas se quisermos resolver outro tipo de problema, teremos que escrever outro programa específico. A meta é um programa geral de soluções, que seja capaz de encontrar soluções para diferentes tipos de problemas. Tal programa seria capaz de demonstrar teoremas matemáticos, resolver problemas de Xadrez ou mesmo resolver palavras cruzadas.
• *Reconhecimento de imagem:* É de importância fundamental na robótica onde, por exemplo, uma mão mecânica controlada por computador pode distinguir entre porcas, parafusos e arruelas — uma utilização bem melhor, pois para um computador, ver por meio de uma câmera de TV e reconhecer o objeto focalizado por ela, pode ser uma tarefa bem difícil para ele.
• *Programação automática:* O operador apenas supriria o computador com especificações do programa, e este se encarregaria de elaborar o programa. Ou então, o computador poderia testar e corrigir programas escritos por pessoas.

Neste artigo, tentaremos mostrar primariamente como escrever programas de jogos inteligentes em BASIC. O BASIC utilizado, será o do TK-82C, 83 ou 85. Embora esta não seja a linguagem ideal para programas de IA, a sua vantagem está no fato de que a grande maioria dos microcomputadores a utiliza como linguagem padrão.

O programador se defrontará com três problemas principais: representações, busca de solução e abrangência. A maneira como superar estes problemas, no decurso da transformação de sua idéia em programa, irá determinar, em grande parte, quão "inteligente" será seu programa.

O problema de como representar uma estrutura de dados complexa em uma linguagem de computador, não é limitado a IA. Entretanto, representação pode ser um problema particularmente difícil para quem está envolvido com IA, uma vez que as estruturas e processos que se tenta modelar, por computador, são bastante complexos.

O problema da busca de solução é crucial em IA. Frequentemente, um programa de IA tenta resolver uma tarefa, gerando uma infinidade de possíveis soluções, testando cada uma, para determinar qual é a correta. Isto é análogo ao estudante que, ao invés de procurar resolver uma equação do segundo grau pela fórmula, meramente atribuisse diferentes valores para X, até encontrar aquele que fosse a solução. O problema é que o conjunto das possíveis soluções, pode ser tão grande, que o computador pode ter dificuldades em gerar e testar todas elas em tempo hábil. Imaginem o estudante que provavelmente testaria milhões de valores de ***X*** até encontrar o correto! (A despeito do fato do método resolver qualquer tipo de equação, não parece muito prático).

Programas de jogos são particularmente vulneráveis ao problema da busca, uma vez que pode haver um número exponencialmente crescente de possíveis movimentos a serem testados, até que o melhor seja encontrado.

Finalmente, há o problema da abrangência. Muitos programas de IA hoje em dia, operam com certos limites estreitos e bem definidos. Embora funcionem bem dentro desses limites, sua utilidade é restrita justamente devido a estes limites. Mas quando o programa é estendido para operar com mais tipos de dados, provido com mais funções e utilitários, surge um efeito negativo: o programa que agora pode realizar um maior número de tarefas, já não as realiza tão bem individualmente.

Em outras palavras, em IA a abrangência (ou generalidades) de um programa é inversamente proporcional a sua eficiência.

PROGRAMAS QUE JOGAM

Em meados da década de 60 o filósofo ***Hubert Dreyfus*** de ***Berkeley*** declarou que "um computador não jogaria xadrez senão em nível de principiante" — pouco depois deste comentário, foi derrotado pelo programa Machac. Dreyfus ficou surpreso, pois acreditava que jogar xadrez exigia certas características humanas, como por exemplo, intuição e criatividade. Seu ponto de vista falha — quando características humanas como criatividade, analisadas em seus constituintes básicos, se revelam como simples sequências de eventos fisiológicos que podem ser imitados (e talvez melhorados) por um programa de computador. Muitos dos programas que têm sido desenvolvidos, tentam jogar Xadrez, Damas ou Go imitando o processo que se desenvolve na mente humana — quando esta se ocupa de um jogo dessa natureza. Muitos destes programas incorporam funções de avaliação que, embora representem o processo de pensamento humano em algum grau, não são modelos deliberados da mente humana.

O programa que teve até hoje o melhor desempenho contra um bom jogador é o ***BKG 9.8*** de Hans Berliner, que joga Gamão. Em um "match" contra o campeão mundial de Gamão, Luigi Villa, ***BKG*** venceu quatro em cinco partidas.

O mais forte programa de Xadrez é o →

CHESS 4.7 da Northwestern University, que ganhou uma partida do grande mestre internacional David Levy em um torneio cujo resultado foi 3,5 a 1,5 (três vitórias, um empate e uma derrota) a favor de Levy.

A FUNÇÃO DE AVALIAÇÃO

Como proceder para escrever um programa de Xadrez ou Damas? O primeiro passo é desenvolver um procedimento exato e preciso, para determinar quando o computador vai jogar e qual movimento é o mais adequado naquele instante de jogo. Este procedimento é chamado função de avaliação. Uma função de avaliação para xadrez ou outro jogo, deve ser específica o suficiente para que possa ser traduzida em um programa de computador. Algumas teorias de estratégia ou fatores posicionais podem ser impossíveis de serem incorporados em um programa, sem um refinamento da teoria, em termos mais precisos.

Dois métodos de programação comumente usados em funções de avaliação, são os de mini-maximização e o algoritmo alfa-beta.

ÁRVORES E MINI-MAXIMIZAÇÃO

Suponhamos que uma pessoa esteja jogando Xadrez com um computador. Esta acabou de mover e passa a vez ao computador. O que o programa fará?

Uma maneira lógica de proceder é construir uma árvore dos possíveis movimentos. O programa verifica — digamos, — seis jogadas à frente: três brancas e três pretas. Dentro desta limitação de seis jogadas, o computador gera todas as possíveis combinações de movimentos. Além disso, para cada movimento possível, o computador associa um valor numérico à configuração resultante no tabuleiro; este valor numérico indica quem está em vantagem, baseado em fatores como material e posição. Para gerar a árvore, o programa sempre assume, que o oponente fará o melhor jogada possível.

Vejamos um exemplo de critério de avaliação de posição para Xadrez:

	Pontos
1. Por aumentar a mobilidade da dama	+ 2
2. Por aumentar a mobilidade do bispo	+ 2,2
3. Por aumentar a mobilidade do cavalo	+ 0,3
4. Por não haver realizado o roque	− 0,4
5. Por estar defendido o peão	+ 0,1

Este tipo de avaliação concede mais pontos à defesa que ao ataque, portanto são necessários os seguintes fatores de correção:

	Pontos
1. Por aumentar a mobilidade do cavalo da dama	+ 0,8
2. Por melhorar a defesa dos peões	+ 0,5
3. Por aumentar a mobilidade da torre da dama	+ 1,0
4. O bispo da dama defende os peões	+ 0,5
5. O peão do rei está defendido pelo cavalo	+ 0,3

Esta avaliação só pode ser realizada em posições "estáticas", ou seja, sem troca de peças ou xeques.

Quando o computador houver gerado a árvore inteira, ele retorna pelo ramo da árvore que resultou em maior vantagem (maior pontuação). Esta técnica é chamada mini-maximização, uma vez que o computador joga para obter a maior vantagem, supondo sempre que o adversário jogará, para lhe dar a menor vantagem possível. Quando o computador determinar qual sequência é a melhor, este executa o primeiro movimento da sequência.

ALGORITMO ALFA-BETA

A utilização de árvores e mini-maximização parece ser a melhor maneira de gerar todos os movimentos possíveis em um jogo. Uma maneira evidente de aumentarmos a força do programa seria verificar mais jogadas à frente e, em lugar de seis movimentos, poderíamos pesquisar 15 movimentos. Se o computador puder fazer isto, provavelmente será imbatível.

A dificuldade em criar uma árvore para 15 movimentos, é que o número de

combinações possíveis é da ordem de 10^{15790}! (Considerando que o universo tem "apenas" 10^{80} átomos, chamar aquele número de astronômico, seria incorreto e injusto).

Evidentemente a quase totalidade destas combinações são desdobramentos de jogadas implausíveis ou fracas, abrindo assim, espaço para redução deste número.

Uma maneira de limitar a quantidade de buscas, é por meio do algoritmo alfa-beta. A idéia por trás dele é simples. Quando o computador está gerando a árvore dos possíveis movimentos, não é necessário gerar o ramo que resulta de um movimento inferior da parte do computador. Em outras palavras, se o computador avaliou o movimento e concluiu que este não é bom, não é necessário verificar os movimentos subsequentes, de modo que o movimento ***X*** não será realizado em hipótese alguma. Analogamente — seguindo a idéia do mini-max — o computador não necessita gerar o ramo da árvore, que resulta de um movimento inferior por parte do oponente.

Quando o algoritmo alfa-beta é empregado como parte de uma busca em árvore, o tempo economizado é fenomenal. Estima-se que em condições ótimas, uma busca de quatro movimentos aumenta sua velocidade em cerca de 500 vezes.

Vejamos alguns critérios que podem nos orientar na eliminação de movimentos (geralmente se utilizam mais de cinquenta critérios para uma avaliação definitiva):

1. Equilíbrio de material
2. Valor das peças
3. Estrutura dos peões
4. Segurança do próprio rei
5. Controle do centro

DAMAS

Vejamos a seguir, a título de exemplo, um programa que joga ***Damas.*** Acompanhe a descrição pelo fluxograma mostrado adiante. No interesse da brevidade, ele não incorpora uma rotina de geração de árvore para os movimentos, como descrito anteriormente. Apenas verifica um ou dois movimentos, dependendo da situação.

O programa está dividido em seis subrotinas — num programa supervisor — que as chama à medida que são necessárias.

A primeira subrotina é a da linha 450. Ela desenha e numera o tabuleiro.

Em seguida é chamada a subrotina da linha 230. Esta rotina coloca as peças em sua configuração inicial. A disposição das peças é representada como uma saída gráfica no vídeo e como um ***array*** 12 x 12 (T$). Necessitamos de um ***array*** 12 x 12, pois se utilizassemos um 8 x 8 certos argumentos da variável **T$** se tornariam negativos em certas operações, o que resultaria em erro de software; deste modo, criamos então, uma margem em torno do tabuleiro.

A subrotina que escolhe o primeiro movimento é a da linha 970. O computador joga sempre com as pretas. Como, em Damas, as pretas sempre iniciam o jogo, o computador sempre começa jogando e isto simplifica um bocado o nosso programa!

A subrotina simplesmente escolhe aleatóriamente um movimento em direção às casas centrais. A subrotina de entrada (linha 780) permite ao jogador introduzir sua jogada. Como a função INKEY$ é utilizada, não há necessidade da utilização do NEWLINE. Assim que o quinto caracter é introduzido, a rotina é terminada.

Os dois primeiros caracteres são as coordenadas da casa de saída, o caracter do meio é um "—" (hífen) para jogadas normais, ou um "x" para saltos duplos ou triplos (capturas) e os caracteres finais são as coordenadas da casa de chegada.

Como o objetivo deste programa é simplesmente ilustrar a mecânica de um programa de jogo, não houve compromissos com relação aos detalhes. Deste modo não há crítica de entrada e é possível a introdução de movimentos ilegais. Além disso uma possível dama não é identificada pelo programa, mas o operador poderá simular seu movimento usando o "x" no lugar do hífen. Como a captura só é reconhecida em jogadas normais, é necessário →

INÍCIO
DESENHA E NUMERA O TABULEIRO 450
CONFIGU-RAÇÃO INICIAL 230
PRIMEIRO MOVIMENTO 970
GOSUB 840
1

```
0 REM "■■■■ DAMAS ■■■■"
1 REM PROGRAMA PRINCIPAL
2 FAST
10 GOSUB 450
20 GOSUB 230
25 SLOW
30 GOSUB 970
35 GOSUB 840
40 GOSUB 780
45 GOSUB 840
50 IF M=1 THEN GOTO 40
55 GOSUB 1040
60 GOSUB 840
65 GOTO 40
230 REM ######################
231 REM CONFIGURACAO INICIAL
235 DIM T$(12,12)
240 FOR I=2 TO 11
250 LET T$(I,2)="X"
260 LET T$(I,11)="X"
270 LET T$(2,I)="X"
280 LET T$(11,I)="X"
290 NEXT I
320 FOR I=3 TO 9 STEP 2
325 LET T$(I,3)="B"
330 LET T$(I+1,4)="B"
335 LET T$(I,5)="B"
340 LET T$(I+1,8)="P"
345 LET T$(I,9)="P"
350 LET T$(I+1,10)="P"
355 PRINT AT 22,3*I;"*"
360 PRINT AT 19,3*I+3;"*"
365 PRINT AT 16,3*I;"*"
370 PRINT AT 7,3*I+3;"■"
375 PRINT AT 4,3*I;"■"
380 PRINT AT 1,3*I+3;"■"
390 NEXT I
395 PRINT AT 0,0;"HUMANO"
396 PRINT AT 4,0;" MICRO"
397 PRINT AT 21,0;"■JOGO■"
400 RETURN
450 REM ######################
451 REM DESENHA E NUMERA O TABU
LEIRO
455 POKE 16418,0
460 LET A$="      ███   ███
███   ███   "
470 LET B$="      ███   ███
█   ███   ███"
475 RAND
480 FOR I=1 TO 4
490 PRINT B$;B$;B$
500 PRINT A$;A$;A$
510 NEXT I
520 FOR I=1 TO 8 STEP 2
530 PRINT AT 0,3*I+7;CHR$ (I+37
)
535 PRINT AT 0,3*I+10;CHR$ (I+1
66)
540 PRINT AT 3*I-1,8;CHR$ (37-I
);AT 3*I+2,8;CHR$ (164-I)
550 NEXT I
560 RETURN
780 REM ######################
781 REM    ENTRADA
785 PRINT AT 20,0;"■HUMANO■"
790 LET K=0
791 LET C$="*B"
795 LET E$=" "
800 IF INKEY$<>"" THEN GOTO 800
810 IF INKEY$="" THEN GOTO 810
815 LET E$=E$+INKEY$
820 PRINT AT 1,0;E$;" "
825 IF LEN E$=6 THEN GOTO 835
830 GOTO 800
835 PRINT AT 20,0;"■MICRO■"
836 RETURN
840 REM ######################
841 REM ALTERACAO DO TABULEIRO
845 IF C$(2)="P" THEN PRINT AT
5,0;E$
850 LET I=CODE E$(2)-35
852 LET J=VAL E$(3)+2
854 LET K=CODE E$(5)-35
856 LET L=VAL E$(6)+2
858 LET M=(E$(4)="X")
860 PRINT AT 31-3*J,3*I;"■"
865 PRINT AT 31-3*L,3*K;C$(1)
870 LET T$(I,J)=" "
875 LET T$(K,L)=C$(2)
880 IF L-J<>2 THEN RETURN
890 PRINT AT 31-1.5*(J+L),1.5*(
I+K);"■"
895 LET T$((I+K)/2,(J+L)/2)=" "
900 RETURN
950 REM ######################
970 REM PRIMEIRO MOVIMENTO
975 LET C$="■P"
980 LET R=INT (RND*4+.5)
985 GOTO (R*5+990)
990 GOTO 980
995 LET E$=" B6-C5"
997 GOTO 1015
1000 LET E$=" D6-C5"
1001 GOTO 1015
1005 LET E$=" D6-E5"
1006 GOTO 1015
1010 LET E$=" F6-E5"
1015 GOSUB 840
1020 RETURN
1040 REM ######################
1041 REM AVALIACAO
1042 REM PARTE A.
1050 LET C$="■P"
1060 FOR I=3 TO 10
1070 FOR J=3 TO 10
1080 IF T$(I,J)<>"P" THEN GOTO 1
110
1090 IF T$(I+1,J-1)<>"B" OR T$(I
+2,J-2)<>" " THEN GOTO 1100
1091 LET T$(I+1,J-1)=" "
1092 PRINT AT 34-3*J,3*I+3;"■"
1093 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (37+I)+CHR$ (J+
24)
1095 RETURN
1100 IF T$(I-1,J-1)<>"B" OR T$(I
-2,J-2)<>" " THEN GOTO 1110
1101 LET T$(I-1,J-1)=" "
1102 PRINT AT 34-3*J,3*I-3;"■"
1103 LET E$=" "+CHR$ (35+I)+CHR$
 (26+J)+"-"+CHR$ (33+I)+CHR$ (J+
24)
1105 RETURN
1110 NEXT J
1111 NEXT I
1120 REM PARTE B.
1130 FOR I=3 TO 10
1140 FOR J=3 TO 10
1150 IF T$(I,J)<>"P" THEN GOTO 1
220
1160 IF T$(I-1,J-1)<>"B" OR T$(I
+1,J+1)<>" " THEN GOTO 1180
1165 IF T$(I,J+2)<>"P" THEN GOTO
 1170
1166 LET E$=" "+CHR$ (35+1)+CHR$
 (J+28)+"-"+CHR$ (I+36)+CHR$ (J+
27)
1167 RETURN
1170 IF T$(I+2,J+2)<>"P" THEN GO
TO 1200
1173 LET E$=" "+CHR$ (37+I)+CHR$
 (J+28)+"-"+CHR$ (I+36)+CHR$ (J+
27)
1175 RETURN
1180 IF T$(I+1,J-1)<>"B" OR T$(I
-1,J+1)<>" " THEN GOTO 1220
1185 IF T$(I-2,J+2)<>"P" THEN GO
TO 1190
1186 LET E$=" "+CHR$ (33+I)+CHR$
 (J+28)+"-"+CHR$ (I+34)+CHR$ (J+
27)
1188 RETURN
1190 IF T$(I,J+2)<>"P" THEN GOTO
 1210
1195 LET E$=" "+CHR$ (35+I)+CHR$
 (J+28)+"-"+CHR$ (I+34)+CHR$ (J+
27)
1197 RETURN
1200 IF T$(I+1,J-1)<>" " THEN GO
TO 1220
1205 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+36)+CHR$ (J+
25)
1207 RETURN
1210 IF T$(I-1,J-1)<>" " THEN GO
TO 1220
1215 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+34)+CHR$ (J+
25)
1217 RETURN
1220 NEXT J
1221 NEXT I
1230 REM PARTE C.
1240 FOR I=3 TO 10
1250 FOR J=3 TO 10
1255 IF T$(I,J)<>"P" THEN GOTO 1
340
1260 IF T$(I-1,J-1)<>" " OR T$(I
-2,J-2)="B" THEN GOTO 1280
1270 IF T$(I-2,J)=" " AND T$(I,J
-2)="B" THEN GOTO 1280
1273 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+34)+CHR$ (J+
25)
1275 RETURN
1280 IF T$(I+1,J-1)<>" " OR T$(I
+2,J-2)="B" THEN GOTO 1340
1290 IF T$(I+2,J)=" " AND T$(I,J
-2)="B" THEN GOTO 1340
1300 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+36)+CHR$ (J+
25)
1310 RETURN
1340 NEXT J
1350 NEXT I
1400 REM QUALQUER MOVIMENTO POSS
IVEL
1410 FOR I=3 TO 10
1420 FOR J=3 TO 10
1430 IF T$(I,J)<>"P" THEN GOTO 1
500
1440 IF T$(I-1,J-1)<>" " THEN GO
TO 1470
1450 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+34)+CHR$ (J+
25)
1460 RETURN
1470 IF T$(I+1,J-1)<>" " THEN GO
TO 1500
1480 LET E$=" "+CHR$ (35+I)+CHR$
 (J+26)+"-"+CHR$ (I+36)+CHR$ (J+
25)
1490 RETURN
1500 NEXT J
1510 NEXT I
1520 STOP
```

fazer com que a dama ocupe a casa que contém a peça preta, e em seguida se mova para a casa desejada.

A subrotina da linha 840 é chamada após cada movimento. Ela altera o ***array*** e o ***display*** para refletir a nova configuração.

ROTINA DE AVALIAÇÃO

A última subrotina (linha 1040) é a maior e mais importante. É a rotina de avaliação, na qual o computador (pretas) decide que movimento executar. A rotina é baseada no seguinte algoritmo:

1. Se as pretas estão em posição de capturar alguma peça branca, então que o façam.

2. Se as pretas estão ameaçadas de captura pelas brancas então:

2a. se for possível a defesa através do bloqueio da retaguarda então que o façam, senão;

2b. se for possível movimentar a peça ameaçada então que o façam, senão;

3. As pretas fazem o primeiro movimento que não resulte em captura da peça movida. Se todos os movimentos resultarem em captura, uma peça qualquer é movida.

Como você pode ver, este é um algoritmo simples e descomplicado que serve para manter as pretas longe dos problemas. Isto resulta em um programa que não é particularmente agressivo, mas que usualmente se defende adequadamente.

A rotina de avaliação se divide em três maiores partes. Parte A, na qual as pretas procuram possíveis capturas, corresponde à parte 1 do algoritmo acima. Parte B, na qual as pretas se defendem de capturas, corresponde à parte 2 do algoritmo, e Parte C onde as pretas procuram um movimento que não resulte em captura, correspondente à parte 3. A organização é hierárquica. Se as pretas não encontram movimentos a serem feitos na Parte A, o controle passa para a Parte B. Se nenhum movimento resulta daqui, então o controle passa para a Parte C.

Vamos dissecar a Parte A da subrotina e ver como ela funciona. A Parte A é implementada com dois loops FOR. . . NEXT; cada um variando de 3 a 10, varrendo todo o tabuleiro.

A linha 1080 percorre todos os quadrados do tabuleiro procurando por uma peça preta. Quando esta é encontrada, as linhas 1090 e 1100 tomam lugar. Elas procuram por uma situação especial que pode ocorrer em dois dos quatro quadrados adjacentes: a situação em que uma peça branca possa ser capturada.

Vamos assumir que uma peça preta esteja localizada em T$(6,6). Esta situação é ilustrada na Fig. 1.

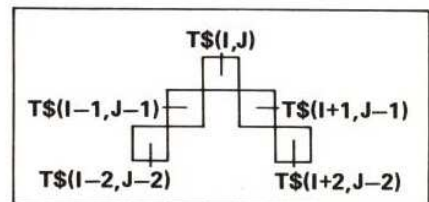



Fig. 1

A linha 1090 procura por uma peça branca em **T$**(I+1,J−1) e por uma casa vazia em **T$**(I+2,J−2). Se ambas as condições são satisfeitas um vazio é colocado em **T$**(I+1,J−1), a subrotina armazena o movimento (salto) em **E$**, chama a subrotina da linha 840 e retorna ao programa principal. Se por outro lado, uma ou ambas as condições não forem satisfeitas, o programa passa para a linha 1100. Esta linha analisará as casas (I−1,J−1) e (I−2,J−2). Apenas quatro casas precisam ser consideradas quando o programa procurar uma possível captura; as casas atrás da peça podem ser ignoradas, uma vez que apenas as damas podem andar para trás e este programa não prevê damas para as pretas. Se não houver captura possível, o controle passa para as Partes ***B*** e ***C*** da subrotina, cujo funcionamento é semelhante a Parte A.

aula 6

Flavio Rossini
Pierluigi Piazzi

Um pouco de lógica: AND, OR, NOT

Existem no TK três "palavras" que podem ser muito úteis quando associadas ao IF; são elas:

AND (tecla 2)
OR (tecla W)
NOT (tecla N Function)

AND significa "e", **NOT** significa "não", enquanto **OR** significa "ou". Suponha que no programa que imprimia "beleza" ou "alegria" mudássemos as condições para: "se o número estiver entre 3Ø e 7Ø, imprima "beleza"; deveremos substituir a linha 5Ø.

```
5Ø IF N>3Ø AND N<7Ø THEN GOTO
1ØØ
```

Para ilustrar as outras duas, façamos um pequeno programa: →

Note que temos em jogo o seguinte conjunto de números possíveis:

[Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1Ø]

Para X < 2 ou X > 8 teremos ⇒[Ø, 1, 9, 1Ø"]

Para NÃO (X < 6) ≡ X ⩾ 6 ⇒[6, 7, 8, 9, 1Ø]

Para X > 3 e X < 7 ⇒[4, 5, 6]

Em BASIC teríamos:

```
  5 SLOW
 10 RAND
 20 FOR I=1 TO 7
 30 LET X=INT (11*RND)
 40 IF X<2 OR X>8 THEN PRINT "L
UZ"
 50 IF NOT X<6 THEN PRINT "SOL"
 60 IF X>3 AND X<7 THEN PRINT "
ESTRELA"
 70 PRINT X
 80 NEXT I
 85 FAST
```

Estas palavras aliadas ao IF serão discutidas novamente em aulas futuras. Note que temos os símbolos <= menor ou igual, (tecla **R**) e >= maior ou igual (tecla **Y**) que também podem ser utilizados com a instrução **IF**.

Vamos agora concluir todos os símbolos "vermelhos" (ou amarelos) do TK com exceção do $ que será visto futuramente; faltam apenas:

:
?
£
STOP (tecla A)
LPRINT (tecla S)
LLIST (tecla G)

Os três primeiros, não têm nenhuma função especial e servem apenas para participarem de "mensagens" que devem estar ***sempre*** entre aspas. (Pode-se acrescentar a eles o **.** que está na mesma tecla que a **,**).

Já o STOP serve apenas pára "quebrar" a execução de um programa: um programa pode constar de várias partes, cada uma terminada por STOP. Ao chegar no STOP o programa para, podendo ser reiniciado a partir da instrução seguinte, usando **CONT** (veja a aula 3). Por exemplo:

```
  5 SLOW
 10 PRINT TAB 5;"BEATLES"
 20 STOP
 30 PRINT TAB 5;"IS LOVE"
 40 STOP
 50 PRINT TAB 5;"FOREVER"
 60 STOP
 70 PRINT
 80 LET I=2
 90 PRINT TAB 7;I
100 IF I=256 THEN STOP
110 LET I=I*I
120 GOTO 90
```

Digite RUN e NEW LINE; a seguir você terá que digitar **CONT** e **NEW LINE** nas primeiras três vezes que o programa parar. O que acontecerá se você fizer **CONT** e **NEW LINE** quando o programa parar pela quarta vez.

Note que ao passar por STOP, o TK imprime a seguinte mensagem no canto esquerdo inferior:

9/ (linha que parou o programa)

Observação: O **STOP** já havia sido memorizado suscintamente na aula nº 4, quando falamos sobre o **INPUT**.

Quanto ao **LPRINT** e ao **LLIST** eles fazem o mesmo que os já conhecidos **PRINT** e **LIST,** só que, ao invés de escreverem na tela de TV, escrevem na impressora. Ainda para lidar com a impressora, existe a "key-word" COPY (tecla **Z**), que copia exatamente o que está desenhado na tela. (Naturalmente você pode fazer um **BREAK** durante um COPY — se não desejar copiar a tela toda).

Algumas observações são necessárias:

a) o **LLIST** permite, assim como o **LIST,** fazer uma listagem a partir de qualquer linha do programa.

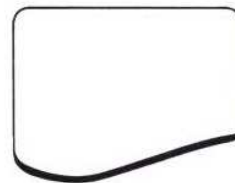
b) Para programas "menores" que uma tela, é conveniente uma COPY ao invés de **LLIST,** pois ele é mais rápido.

c) Não existe a instrução **LPRINT AT** (tente executá-la . . .), mas existe a instrução **LPRINT TAB**.

d) Com exceção do **AT** e do **SCROLL**, o **PRINT** e o **LPRINT** são perfeitamente compatíveis com relação ao posicionamento da impressão (**TAB, ;**); no entanto, se você for utilizar números na faixa de Ø,ØØØØ1 a Ø,ØØ99999999 a instrução não funciona adequadamente, experimente:

```
10 LET X=0.00001
20 LPRINT X
```

Em aulas futuras, veremos como contornar este problema.

O **LPRINT** tem um símbolo especial para diagrama de blocos:

O que acontece se tentarmos usar estes comandos sem a impressora estar conectada?

Exercícios:

1. Faça um programa capaz de calcular o fatorial de um número inteiro positivo. O programa ***não*** deve aceitar números negativos e/ou fracionários imprimindo uma mensagem de erro.

Define-se fatorial de ***n,*** como sendo:

n! = n.(n−1).(n−2). . . 1

Assim

5! = 5 . 4 . 3 . 2 . 1 = 12Ø

2) Escreva um programa que leia o número e mostre, na tela, se o número é par ou ímpar. Seguindo o mesmo raciocínio, faça então outro programa que escreva na tela "múltiplo de 6" toda vez que o número for múltiplo de 6 — (Dica: $a \div b = c$ com resto d; como se calcula d?)

CARACTERÍSTICAS TÉCNICAS:

UNIDADE DE PROCESSAMENTO:
- Microprocessador Z80 A (8 bits) velocidade 3,25 MHz.

MEMÓRIAS:
— 8 Kbytes EPROM (sistema interpretador BASIC).
— 16 Kbytes RAM (utilização para o usuário).

MONITOR DE VÍDEO:
— TV B&P ou côr no canal 2 (sem modificação)
— 24 linhas com 32 caracteres por linha.
— Gráficos 44 x 64 pixels.
— Vídeo reverso e normal comutável por chave

SOFTWARE:
— Compatível com: CP200, TK82-C, TK83, TK85, NEZ8000, TIMEX 1000, ZX80, ZX81.
— Programável em linguagem de máquina.
— Basic residente

UNIDADE DE FITA:
— Um gravador (sem interface AS-100)
— Usando interface AS-100: até quatro gravadores.

TECLADO:
— Membrana com ação sensitiva de toque (bolha metálica inoxidável).
— 40 teclas com 154 funções, caracteres alfanuméricos, caracteres gráficos, funções matemáticas, comandos.

EXPANSÃO:
— Para 32 Kbytes e 48 Kbytes (64 Kbytes total.)
— Adiciona-se um ou dois módulos de 16 Kbytes, no interior do micro, através de acesso próprio no fundo do mesmo.
— O módulo básico de 16 Kbytes, e os dois de expansão, são todos iguais, permitindo expansão progressiva pelo usuário.

ACESSÓRIOS:
— Dois cabos para conexão de gravador.
— Um cabo para conexão do micro a TV.
— A fonte de alimentação (110/220V) já é embutida no AS-1000 dispensando qualquer fonte externa.

PERIFÉRICOS:
— Modem para transmissão de dados.
— Joystick.
— AS-100 (interface):
- opera e gerencia até quatro gravadores.
- comanda qualquer impressora (inclusive profissionais).
- manipula arquivos em fita e não somente programas inteiros.
- avança e pára gravadores, também por software (programação).
- testa validade de programas e arquivos em fita.
- permite cópias de fitas.
- lista programas, dados e arquivos na impressora.
- exclusivo sistema FSK (frequency shift keying) com compressor de áudio, que ajusta automaticamente o nível para o gravador, eliminando a desagradável e crítica tarefa de ajuste no mesmo, oferecendo total confiabilidade nas gravações em fita.
— Módulos de expansão (vide memórias).
— Speed file:
- expansão externa de memória com 1, 2 ou 4 Mbytes.

GARANTIA DE 1 ANO
ASSISTÊNCIA TÉCNICA DE FÁBRICA

ENGEBRÁS

ENGEBRÁS ELETRÔNICA E INFORMÁTICA LTDA.
ADMINISTRAÇÃO DE VENDAS: Rua Russel, 450 - 3.º - Tel.: 205-4898
CEP 22210 - RIO DE JANEIRO - RJ

PEQUENOS ANÚNCIOS

Troco, vendo, compro **PROGRAMAS PARA TK, CP 200, NE, TIMEX, SINCLAIR.** Fone: 521-5103 ou escreva para obter relação de programas disponíveis. *José Bisioli Sº. Rua Dr. Samuel de Castro Neves, 72 - apto. 33 – SP - CEP 04726.*

Vendo **PROGRAMAS PARA TK-82C EM LINGUAGEM DE MÁQUINA.** Vários preços ou fita com 10, por Cr$ 30.000,00. Tenho uma lista que inclui: Krazy Kong, Xadrez, Labirinto e muitos outros. Contatar: *José Antonio, Rua Conde de Bonfim, 912/501. Tijuca, RJ. CEP 20530, ou pelo telefone: (021) 258-4537.*

VENDO CP-200 com 12 fitas incluindo aplicativos, ainda na garantia. Valor: Cr$ 200.000,00. Tratar com *Maurício Capeleiro, no telefone (081) 341-4779 ou pela Caixa Postal 4068, Recife, PE.*

Vendo **TK-82 COM EXPANSÃO DE 16K** ou troco por TK-85. *Tita Movadab – telefone: (011) 825-0866 (SP)* a partir das 19 hs.

Se seu problema é **PROGRAMA PARA SEU TK,** procure-me. Tenho o programa que você quer. Informações com *Maurício Xavier, no telefone 493.3322 – Av. Maria Cunha Aguiar, Rua Particular, 68 – Santo Amaro.*

Troco ou vendo **PROGRAMAS** para os micros **ZX-SPECTRUM/81; TK-82/85 e CP200.** Possuo o Galactica, kons, Air-Defense, Combat e muitos outros. Os interessados devem comunicar-se com *Eduardo Medeiros. Rua Eliseu Guilherme, 1076 – Sumaré. Ribeirão Preto, SP – CEP 14100 ou pelo Tel. (016) 625-5796.*

Vendo **NEZ8000, EXPANSÃO 16K,** pouco uso. Livro "45 Programas para TK82-C e NEZ8000". *Fone: (021) 393.9523, à noite com Rainer Schatzmayr.*

Gostaria de entrar em contato com **USUÁRIOS DO TK,** que tenham interesse em programas na área médica. *Luiz Alberto Loss. R. J. de Castilhos, s/nº. Nova Roma – 95250 – Antonio Prado, RS.*

SINCLAIR CLUB – Gostaríamos de intercambiar programas, livros, dicas de programação e assuntos afins, com usuários dos sistemas SINCLAIR e similares compatíveis (TK-82, 83 e 85; CP-CP-200; ZX 81, etc.). Cartas para *SINCLAIR CLUB – Caixa Postal 191, São Bernardo do Campo, SP. CEP 09700.* Envie Cr$ 45,00 em sêlos para receber lista de programas.

Compro **PROGRAMAS PARA TK 82-C,** especialmente jogos (em fitas ou em listagens). Interessados, enviar relação dos programas disponíveis para: *Roberto Gibello. Rua Cel. Joviniano Brandão, 459 – Moóca, SP. CEP 03127.*

Desejo **TROCAR PROGRAMAS** (Jogos, Sub-rotinas, Dicas, etc, para TK e CP 200). *Telefone (011) 240-1883, falar com Ernesto Mathiason.*

Troco **PROGRAMAS ESCRITOS** para TK-82C (Basic ou Assembly), principalmente jogos. *Claudio Alves de Lima. Rua Chicago, 33 – 04564, São Paulo, SP.*

RADIOAMADOR. Vendo Interface e programa para decodificar telegrafia para o seu TK-82C, TK-85 e Sinclair ZX81. Informações com *Renato Strauss. PY2-EMI. Rua Cardoso de Almeida, 654/32 – 05013, São Paulo, SP.*

VD TK82-C mais 16 K, Joystick e fitas com 40 jogos importados. Tratar com *Fernando A. Morales Castro no telefone: 63-2155.*

QUEBRA-CABEÇA

Este mês: O CAPITÃO MACHISTA

O problema proposto a seguir é uma variante do *Problema de Josephus*. Diz uma estória, que Josephus estava com outros 40 judeus, em uma caverna e que deveriam todos morrer para não serem jogados aos leões pelos romanos. Como ele não queria morrer de nenhum modo, colocou todos os judeus em um enorme círculo e estipulou que contaria, de três em três, a partir dos judeus. Aqueles em que fosse caindo esse número, seriam mortos. Josephus colocou-se na 16ª posição e seu melhor amigo na 31ª posição. Dos 41 judeus existentes, restaram apenas os dois. O problema abaixo, é bem mais simples que o de Josephus.

"Num barco, estão 6 mulheres e 7 homens, sendo o comandante um dos homens. Devido a um vazamento, o comandante chega à conclusão de que para o barco não afundar, nele deverão permanecer apenas 7 pessoas. Em outras palavras, seis deverão servir de comida aos tubarões.

Para evitar problemas com a escolha das pessoas que seriam sacrificadas, uma das mulheres, ferrenha feminista, propôs que os passageiros fossem dispostos numa circunferência, sendo que, ao lado de cada mulher, haveria sempre dois homens. O capitão escolheria um número e começaria a contar no círculo, a partir de uma das pessoas. Aquele que caisse no número escolhido, seria jogado ao mar e a contagem do mesmo número iniciar-se-ia a partir da pessoa seguinte.

O capitão, um ferrenho machista, pensou um pouco, fêz um rápido programa em seu TK 83, ocupando apenas 1 k Byte de RAM e aceitou a sugestão. Um dos homens, jovem engenheiro, temendo que o capitão escolhesse o número 2 (dois) e iniciasse a contagem por sua mulher, fêz questão que a contagem começasse por ele. O capitão escolheu um número, contou-o por 6 vezes na circunferência — que a cada vez ficava menor e conseguiu ser o único homem em seu barco".

O Quebra-Cabeça consiste em obter um programa para o TK que, ocupando menos de 1 kByte de RAM, encontre o menor número que o capitão pode ter escolhido.

AS IDADES DAS TRÊS FILHAS

Até a data de fechamento da *Microhobby* nº 6, recebemos apenas três soluções do Quebra-Cabeça "As Idades das Três Filhas". Felizmente elas estavam corretas. Esse problema, como já dissemos na edição nº 4, é bem antigo e surpreendeu-nos o fato de recebermos apenas três soluções. Provavelmente, o que muitos leitores não perceberam foi a importância da última dica:

"A mais velha dança ballet".

O essencial desta afirmação está no fato de ***existir uma mais velha.*** O José Hilário, tendo observado muito bem isso, solucionou o problema com menos de 2 kBytes de RAM, por isso escolhemos a sua solução para publicar. Abaixo, transcrevemos sua carta. Recebemos mais duas cartas que nos foram enviadas por: Gilberto Gamer e Carlos Eduardo Carvalho.

"Caros amigos da ***Microhobby***, resolvi encarar o quebra-cabeça proposto por vocês na revista nº 4. A maneira que encontrei para solucioná-lo, foi através da "eliminação"; pois, existem poucas possibilidades de o produto de três números inteiros positivos, ser 36 e depois, à cada pista fornecida ao matemático B, surge uma maneira de descartar algumas destas possibilidades. Como eu tenho confiança na palavra de um matemático (dentro de uma certa lógica, é claro!), nem passou pela minha cabeça a idéia de que esse problema não teria solução. E o programa listado abaixo, mostra isso. Vejam! .

José Hilário da Cruz

```
   1 REM AS IDADES DAS TRES IRMA
S
   2 REM JOSE HILARIO DA CRUZ
   3 FAST
   5 LET M=0
  10 LET N=0
  15 DIM A(9)
  20 DIM B(9)
  30 DIM C(9)
  35 DIM S(9)
  40 FOR I=1 TO 3
  45 FOR J=1 TO 6
  50 LET K=INT (36/(I+J))
  55 IF I*J*K=36 AND (I<=J AND J
<=K) THEN GOSUB 125
  60 NEXT J
  65 NEXT I
  70 FOR I=1 TO M
  75 FOR J=1 TO M
  80 IF S(I)=S(J) AND I<>J THEN
GOSUB 155
  85 NEXT J
  90 NEXT I
  95 FOR I=1 TO N
 100 IF (A(I)>B(I) AND A(I)>C(I)
) OR (B(I)>A(I) AND B(I)>C(I)) O
R (C(I)>A(I) AND C(I)>B(I)) THEN
 GOTO 115
 110 NEXT I
 115 PRINT "COM CERTEZA AS IDADE
S SAO:          ";A(I);" ANOS";",",,
B(I);" ANOS    E",,C(I);" ANOS"
 120 STOP
 125 LET M=M+1
 130 LET S(M)=I+J+K
 135 LET A(M)=I
 140 LET B(M)=J
 145 LET C(M)=K
 150 RETURN
 155 LET N=N+1
 160 LET A(N)=A(I)
 165 LET B(N)=B(I)
 170 LET C(N)=C(I)
 175 RETURN
```

PROGRAMAS DO LEITOR

O ESQUIADOR

Edson Mikio Yoshida
Nelson Murassaki

Este é um programa de 2k, para ser rodado em SLOW utilizando-se um pouco de linguagem de máquina, possibilitando assim, o aumento da velocidade do jogo.

Basicamente, você controla um *Esquiador* (letra H) que vai descendo uma pista de neve e encontra, na sua frente, vários obstáculos. Enquanto ele desce, não é possível alterar sua trajetória, porém ao se encontrar com um obstáculo, ele pára e, a partir daí, podemos escolher para que lado dirigir a trajetória, 5 (para a esquerda) e 8 (para a direita). Note que, se o *Esquiador* não puder se movimentar (se o obstáculo não permitir), ele estará fadado ao cruel destino de ser congelado. Digitando-se a letra "O" aparecerão quantos metros você percorreu.

Último detalhe e aviso: "Se você for em uma direção, não poderá voltar atrás, pois a neve é fofa e o pobre coitado não tem fôlego para andar por cima de neve já pisada."

```
   1 REM E£RND FAST )5 ;SGN "CHR
$ " GOSUB ▣TAN
   2 DIM M$(9,26)
   3 RAND
   4 LET Z=VAL "363"+PEEK 16396+
256*PEEK 16397
   5 CLS
   6 LET S=VAL "0"
   7 LET X=VAL "11"
   8 LET Y=VAL "12"
   9 LET M$(1)="[illegible]"
  10 LET M$(2)="[illegible]"
  11 LET M$(3)="[illegible]"
  12 LET M$(4)="[illegible]"
  13 LET M$(5)="[illegible]"
  14 LET M$(6)="[illegible]"
  15 LET M$(7)="[illegible]"
  16 LET M$(8)="[illegible]"
  17 LET M$(9)="[illegible]"
  18 LET A=INT (RND*9)+PI/PI
  19 LET S=S+PI/PI
  20 PRINT AT X,Y;"H"
  21 LET W=PEEK (Z+34+Y)
  22 IF W<>0 THEN GOTO 26
  23 LET U=USR 16514
  24 PRINT AT 21,0;M$(A)
  25 GOTO 18
  26 LET A$=INKEY$
  27 IF A$="" THEN GOTO 26
  28 IF A$="5" AND PEEK (Z+Y)=0
THEN LET Y=Y-PI/PI
  29 IF A$="8" AND PEEK (Z+2+Y)=
0 THEN LET Y=Y+PI/PI
  30 IF A$="O" THEN GOTO 32
  31 GOTO 18
  32 PRINT AT 0,26;S;AT 1,26;"ME
TROS"
  33 GOTO 4

MEMORIA OCUPADA = 938 BYTES
```

Vamos apresentar aqui algumas sugestões para aumentar a velocidade dos seus programas e economizar espaço de memória.

AUMENTANDO A VELOCIDADE

Planeje seu programa de maneira precisa e eficiente. Escreva cuidadosamente o algoritmo e/ou fluxograma. Estude e ataque cada problema antes de digitar o programa.

Use variáveis no lugar de constantes, quando estas são usadas muitas vezes. O **Applesoft** leva muito mais tempo para converter uma constante para a sua representação interna do que para ler o conteúdo de uma variável.

Use apenas **NEXT**, sem o nome da variável de controle. Assim o computador não tem que verificar se a variável é a correta.

Coloque linhas frequentemente usadas no início do programa, já que o Basic inicia a procura de uma linha do início da memória de programação.

Faça com que seu programa execute, de vez em quando, um **FRE(Ø)**, que apaga as informações desnecessárias da memória do computador, melhorando sensivelmente o tempo de acesso aos dados armazenados em variáveis alfanuméricas.

ECONOMIZANDO MEMÓRIA

Renumere seus programas de 1 em 1, começando pela linha Ø ou 1. Dessa forma, os números nos comandos **GOTO** e **GOSUB**, que são armazenados como cadeias de caracteres, serão menores, usando portanto menos memória.

Já que o **Applesoft** não guarda mais do que as duas primeiras letras do nome de uma variável, não há razão para escrevermos, por exemplo, CONTADOR em vez de CO; use, portanto, nomes de apenas uma letra e, se necessário, de duas.

Utilize os dois pontos (":"), de tal forma que haja menor número de linhas em seu programa. Cada nova linha iniciada no **Applesoft** ocupa 5 bytes de memória, além do conteúdo da própria linha.

Sempre use **DIM**, dimensionando as matrizes de forma a ocuparem o menor espaço possível, e utilizando sempre o elemento de índice zero.

Quando seu programa estiver funcionando bem, apague todos os REMs, que são as linhas de programação que, em geral, ocupam o maior espaço na memória. Guarde sempre uma versão do programa

por dentro do apple

Prof. Wilson José Tucci — Coordenador de Projetos Especiais da Escola Experimental Pueri Domus.

com os REMs, mas use apenas a versão mais curta.

Apague as palavras desnecessárias. Mude, por exemplo, a sequência **THEN GOTO** para simplesmente **THEN** ou **GOTO**. Não use o **LET** na definição de variáveis, nem o nome do arquivo depois de um **CLOSE**, sempre que possível.

Planeje seu programa. Repare que as seguintes seqüências fazem a mesma coisa, mas a segunda utiliza menos memória e é mais rápida:

```
10  IF A < 0 THEN B = 60
20  IF A > = 0 THEN B = 77

10 B = 77: IF A < 0 THEN B = 60
```

DEPURANDO PROGRAMAS EM LINGUAGEM DE MÁQUINA LINHA POR LINHA

Programas em linguagem de máquina são poderosos mas complicados, e por isso programas que venham ajudar a analisá-los ou depurá-los são de grande valia para o programador.

O seguinte programa, adaptado da revista **Nibble,** permite analisar um programa escrito em linguagem de máquina, linha por linha. Apesar do programa ser escrito em **Applesoft**, a parte principal da rotina é executada por um programa em linguagem de máquina; a finalidade do programa **BASIC** é então carregá-lo e preparar para a sua execução.

PONTOS DE INTERESSE NO PROGRAMA

Linhas 100-110: Pedem o endereço inicial do programa a ser analisado, que deve ser dado no sistema hexadecimal.

Linhas 120-130: Verificam se foi digitado um número de 1 até 4 dígitos.

Linhas 140-260: Verificam se há apenas dígitos válidos no número.

Linhas 270-350: Convertem o endereço hexadecimal para o seu equivalente no sistema decimal, já que o Applesoft aceita apenas números decimais.

Linhas 360-420: Carregam a rotina escrita em linguagem de máquina que faz a análise do programa. A rotina está armazenada no comando **DATA** da linha 420, codificada byte por byte em números decimais. A rotina é carregada a partir do endereço guardado em EC, que pode inclusive ser alterado pelo usuário do programa.

Linhas 440-450: Dão instruções e iniciam a execução da rotina.

```
0  REM *** DESMONTADOR ***
@1983 POR DENTRO DO APPLE

100  HOME : POKE 35,21: VTAB 8
110  INPUT "ENDERECO INICIAL EM H
     EX =$";AD$
120 LG =  LEN (AD$)
130  IF LG = 0 OR LG > 4 GOTO 260

140  DIM DI$(4): READ DI$(1),DI$(
     2),DI$(3),DI$(4): DATA  0,0,
     0,0
150 X = 4:Y = LG
160  FOR I = 1 TO LG
170 DI$(X) =  MID$ (AD$,Y,1)
180 X = X - 1:Y = Y - 1
190  NEXT I
200  REM VERIFICA VALIDADE DOS DI
     GITOS - HEX 0 ATE F
210  FOR I = 1 TO 4
220 V =  ASC (DI$(I))
230  IF V < 48 OR V > 70 GOTO 260

240  IF V > 57 AND V < 65 GOTO 26
     0
250  NEXT : GOTO 280
260  PRINT "ERRO NO ENDERECO HEX
     =$";AD$; PRINT "APERTE ALGUM
     A TECLA PARA RECOMECAR ;:GET
      A$:RUN
270  REM CONVERTE HEX PARA DECIMA
     L
280  FOR I = 4 TO 1 STEP  - 1
290 V =  ASC (DI$(I))
300  IF V > 64 THEN V = V - 7
310 V = V - 48
320 DE = DE + (V * (16 ^ P)):P =
     P + 1: NEXT
330  PRINT : PRINT  TAB( 10);"END
     ERECO DECIMAL=";DE
340 HI =  INT (DE / 256)
350 LO = DE - (HI * 256)
360  REM CARREGAR ROTINA EM LINGU
     AGEM DE MAQUINA
370 EC = 36864: REM CARREGA APART
     IR DE EC
380 LB = 06: REM BYTE BAIXO DO EN
     DERECO
390 HB = 07: REM BYTE ALTO DO END
     ERECO
400  POKE 0 + LB,LO: POKE 0 + HB,
     HI
410  FOR I = EC TO EC + 25: READ
     IN: POKE 0 + I,IN: NEXT
420  DATA  165,6,133,58,165,7,133
     ,59,169,1,32,99,254,173,0,19
     2,16,251,141,16,192,201,155,
     208,239,96
430  VTAB 23
440  PRINT "QUALQUER TECLA - LIST
     A (ESC) - PARA": VTAB 21
450  CALL EC
460  POKE 35,23: HOME : END

RUN
ENDERECO INICIAL EM HEX =$9000

       ENDERECO DECIMAL=36864
QUALQUER TECLA - LISTA (ESC) - PARA

9000-   A5 06      LDA   $06
9002-   85 3A      STA   $3A
9004-   A5 07      LDA   $07
9006-   85 3B      STA   $3B
9008-   A9 01      LDA   #$01
900A-   20 63 FE   JSR   $FE63
900D-   AD 00 C0   LDA   $C000
9010-   10 FB      BPL   $900D
9012-   8D 10 C0   STA   $C010
9015-   C9 9B      CMP   #$9B
9017-   D0 EF      BNE   $9008
9019-   60         RTS
```

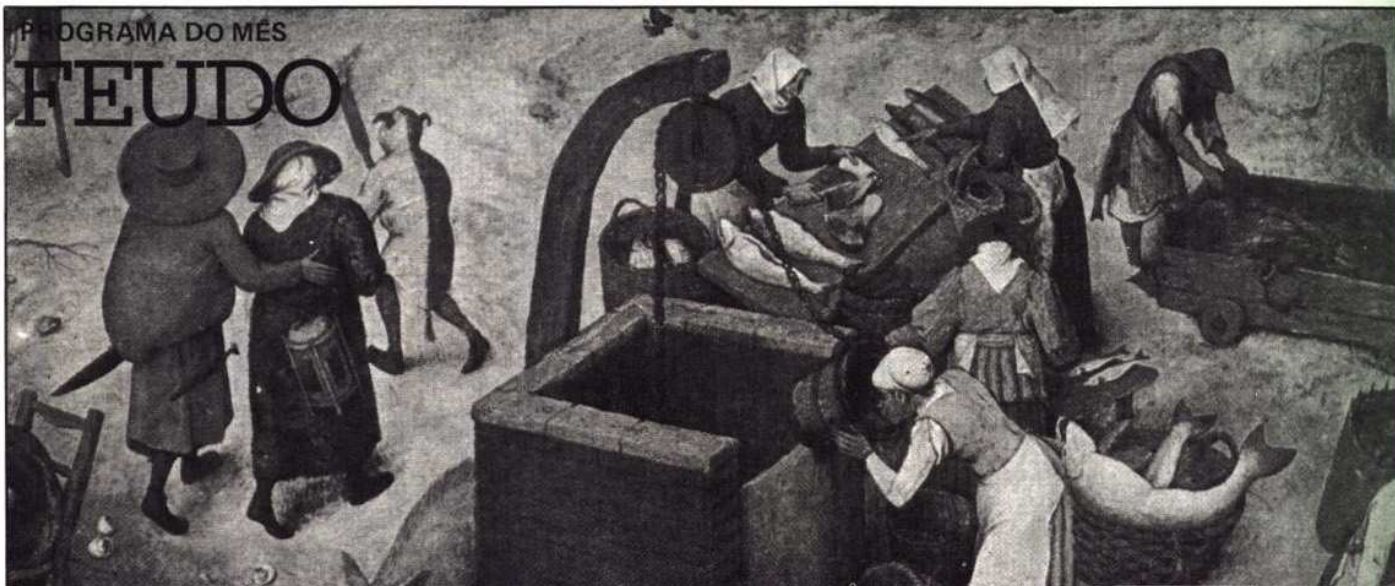
TWO LINERS

Alô pessoal! Apresentamos abaixo a listagem de um **two liners** interessante. Carregue-o no seu computador e verifique o resultado. . . Estamos esperando o seu!

```
1  HGR2 :E =  INT (191 *  RND (1)
     ):D =  INT (279 *  RND (1)):
     C =  INT (3 *  RND (1)) + 2:
      FOR B = 0 TO 191 STEP C: HCOLOR=
      INT (6 *  RND (1)) + 1
2  FOR A = 0 TO 270 STEP C: HPLOT
     D,E TO A,B: NEXT : NEXT : CALL
   62454: FOR T = 1 TO 100:Z =
    PEEK ( - 16336): NEXT : GOTO
    1
```

Enviem suas colaborações para a "Seção Por Dentro do Apple" no seguinte endereço:

MICROMEGA P.M.D.
Rua Bahia, 1049
01244 – São Paulo, SP.

Colaborações de Daniel Falconer e José Eduardo Moreira — professores assistentes do Departamento de Computação da Escola Pueri Domus.

FEUDO

Roberto Bertini Renzetti

Este mês *Microhobby* preocupou-se em possibilitar aos nossos leitores uma oportunidade, se bem que por alguns momentos, de se sentirem donos do poder, senhores de feudos (como um sonho, no retorno à Idade Média) donos de enormes quantidades de terra; a possibilidade de atiçar os instintos "maquiavélicos" de cada um.

Assim, resolvemos escolher o programa "Feudo".

O "Feudo" pode ser disputado por um ou mais jogadores, todos irão, durante três anos, administrar um feudo com enorme quantidade de terra, que possui várias represas (por sinal, bem precárias) e que é cercado por uma floresta onde vivem, escondidos, muitos ladrões.

Um de seus deveres como "senhor feudal" é a preservação de seus domínios e para isso, você deve distribuir seus moradores nas três tarefas básicas (abaixo) existentes:

a. Plantação, cultivo e colheita de trigo.

b. Defesa de seus bens, dos ladrões da floresta.

c. Manutenção das represas, conservando-as em bom estado, ao contrário, seu feudo irá por água abaixo! . . .

Como nem tudo são flores, ocorrem muitos problemas dentro e fora do feudo: muitas pessoas ficam doentes e morrem devido a ataques de bandidos, por fome ou outros motivos e além disso, ocorrem enchentes que inundam e estragam as plantações. Você e outros jogadores como senhores do feudo, são responsáveis pelos moradores e pela resolução dos problemas, para isto, devem ter ações muito bem ponderadas.

O programa, inicialmente, mostra as instruções básicas e, a seguir indaga quantos jogadores irão participar, seus respectivos nomes, idades e sexos.

Colhidos estes dados iniciais surgirá o relatório de cada participante, especificando todos os fatos que ocorreram durante a última estação do ano. (Figura 1).

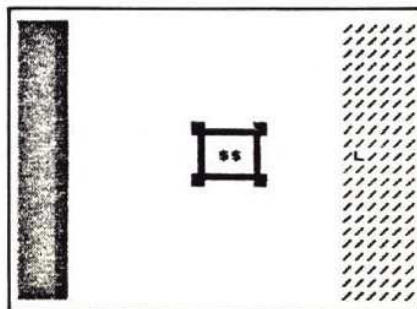
```
          PRIMAVERA ANO 1
BARAO ROBERTO

67 PESSOAS VIERAM AO FEUDO.

          INFORTUNIOS:
 FOME . ENCHENTES . LADROES
   0        0          0
********************************
             VOCE TEM:
             $1000,00
          1067 PESSOAS,
      2500 SACOS DE TRIGO.
********************************
TRABALHAM:200 PESSOAS
(A) REPARO DA REPRESA 200
(B) PLANTAR TRIGO     667
(C) DEFESA DO CASTELO 200
QUANTOS SACOS DE TRIGO SERAO
PLANTADOS?1279

         DIGITE NEW LINE.
```

Observe o seguinte: este relatório servirá para a distribuição dos moradores nas três tarefas básicas já especificadas anteriormente. Para esta distribuição o programa pede qual a quantidade de moradores que irá trabalhar na represa posteriormente, no plantio de trigo e por fim, na defesa da propriedade do "senhor feudal".

É conveniente, na distribuição dos moradores, deixar um número maior de pessoas durante a primavera responsáveis pelo plantio de trigo (pois esta é a estação mais favorável para este plantio) porém, não se deve plantar todo o trigo, afinal o povo também precisa comer!

Depois de efetuado o relatório, o programa apresentará um mapa esquemático de sua propriedade (Figura 2). O lado esquerdo (apresentado na tela) representa as várias represas e, se acaso as comportas baixarem, isto significa que haverá enchentes e consequentemente, morte.

No centro da tela, está seu castelo com seus bens, à direita a floresta – onde moram os ladrões (letra "L") que só pensam em roubá-lo.

O solo de sua propriedade é extremamente fértil e o trigo cresce durante todo o ano, se você replantá-lo e vendê-lo, dele sairá sua riqueza. Lembre-se deste detalhe pois se você, por alguma decisão errada, ficar sem trigo para replantar e vender, o povo sucumbirá à fome já que não haverá mais trabalho, e você não terá dinheiro para comprar comida. A única chance de recuperar este erro é na primavera (a estação do plantio); o programa verá que você não tem trigo e lhe mostrará quanto você terá que comprar de trigo para plantar.

```
VOCE NAO TEM TRIGO SUFICIENTE P/
ALIMENTAR SUA POPULACAO.
VOCE DEVE COMPRAR ALGUM...

VALOR ATUAL DE COMPRA = 18

VOCE TEM $        54905

O MAXIMO QUE VOCE PODE TER SAO
3050 SACOS
QUANTOS VOCE QUER COMPRAR ?
```

Passados três anos, lista-se a relação de jogadores e suas respectivas riquezas angariadas e o vencedor, finalmente, será indicado. Este será o dono de toda a propriedade, sorte dele!

Nota do Editor: Este programa deve ser entendido apenas como uma ***aproximação*** de um feudo, por alguns motivos:

```
   6 CLS
   7 PRINT AT 3,7;"****************"
   8 PRINT AT 4,7;"*              *"
   9 PRINT AT 5,7;"*    FEUDO     *"
  10 PRINT AT 6,7;"*              *"
  20 PRINT AT 7,7;"****************"
  30 PRINT AT 10,0;"VOCE TEM QUE GOVERNAR UM FEUDO";AT 12,0;"NUM PERIODO DE 3 ANOS.VOCE DEVE";AT 14,0;"TENTAR MANTER TODOS VIVOS,PARA"
  40 PRINT AT 16,0;"QUE ELES FACAM UM DOS TRES TRA-";AT 18,0;"BALHOS.DEVE PROTEGE-LOS DAS..."
  50 PRINT AT 21,9;"DIGITE NEW LINE."
  60 IF INKEY$="" THEN GOTO 60
  70 FOR F=10 TO 21
  80 PRINT AT F,0;"                                "
  90 NEXT F
 100 PRINT AT 10,0;"...(A)=>ENCHENTES";AT 11,0;"...(B)=>MORTE PELA FOME";AT 12,0;"...(C)=>LADROES";AT 14,0;"VOCE TEM QUE SER O MAIS RICO";AT 16,0;"POSSIVEL.NO FIM DE 3 ANOS,O MAIS";AT 18,0;"RICO VENCERA..."
 110 PRINT AT 21,9;"DIGITE NEW LINE."
 120 IF INKEY$="" THEN GOTO 120
 130 FOR F=10 TO 21
 140 PRINT AT F,0;"                                "
 150 NEXT F
 160 PRINT AT 10,0;"QUANTOS JOGADORES ? ";
 170 INPUT A
 180 PRINT A
 190 DIM N$(A,11)
 200 DIM U(A)
 210 GOSUB 2360
 220 PRINT AT 21,9;"DIGITE NEW LINE."
 230 IF INKEY$="" THEN GOTO 230
 240 FOR F=1 TO 22
 250 SCROLL
 260 NEXT F
 270 GOSUB 2580
 280 DIM M(A)
 290 DIM C(A)
 300 DIM Q(A)
 310 FOR F=1 TO A
 320 LET M(F)=1000
 330 LET Q(F)=1000
 340 LET C(F)=2500
 350 NEXT F
 360 DIM D(A)
 370 DIM N(A)
 380 DIM Z(A)
 390 DIM X(A)
 400 DIM T(A)
 410 FOR F=1 TO A
 420 LET Z(F)=0
 430 LET X(F)=0
 440 LET T(F)=0
 450 NEXT F
 460 REM ****COMECO****
 470 LET S=0
 480 LET Y=0
 490 LET S=S+1
 500 IF S=1 THEN LET Y$="PRIMAVERA"
 510 IF S=2 THEN LET Y$="VERAO"
 520 IF S=3 THEN LET Y$="OUTONO"
 530 IF S=4 THEN LET Y$="INVERNO"
 540 IF S=1 THEN LET Y=Y+1
 550 IF S=4 THEN LET S=0
 560 FOR P=1 TO A
 570 LET N(P)=INT (RND*70)
 580 LET Q(P)=Q(P)+N(P)
 590 PRINT AT 0,9;Y$;" ANO ";Y
 600 IF N$(P)(11)="M" AND U(P)>18 THEN PRINT "BARAO ";N$(P)( TO 10)
 610 IF N$(P)(11)="F" AND U(P)<=18 THEN PRINT "MARQUESA ";N$(P)( TO 10)
 620 IF N$(P)(11)="M" AND U(P)<=18 THEN PRINT "MARQUES ";N$(P)( TO 10)
 630 IF N$(P)(11)="F" AND U(P)>18 THEN PRINT "BARONESA ";N$(P)( TO 10)
 640 PRINT
 650 PRINT N(P);" PESSOAS VIERAM AO FEUDO."
 660 PRINT
 670 PRINT TAB 9;"INFORTUNIOS:"
 680 PRINT " FOME . ENCHENTES . LADROES"
 690 PRINT TAB 3;X(P);TAB 12;Z(P);TAB 22;T(P)
 700 PRINT "**********************************"
 710 PRINT TAB 12;"VOCE TEM:"
 720 PRINT TAB 15-(((LEN (STR$ M(P)))+2)/2);"$";M(P);",00"
 730 PRINT TAB 15-(((LEN (STR$ Q(P)))+8)/2);Q(P);" PESSOAS,"
 740 PRINT TAB 15-(((LEN (STR$ C(P)))+15)/2);C(P);" SACOS DE TRIGO"
 750 PRINT "**********************************"
 760 PRINT "TRABALHAM:";Q(P);" PESSOAS"
 770 PRINT AT 15,0;"(A) REPARO DA REPRESA ";
 780 INPUT A1
 790 IF INT A1<>A1 THEN GOTO 780
 800 PRINT A1
 810 PRINT AT 14,10;Q(P)-A1;" PESSOAS    "
 820 PRINT AT 16,0;"(B) PLANTAR TRIGO     ";
 830 INPUT A2
 840 IF INT A2<>A2 THEN GOTO 830
 850 PRINT A2
 860 PRINT AT 14,10;Q(P)-A1-A2;" PESSOAS    "
 870 PRINT AT 17,0;"(C) DEFESA DO CASTELO ";
 880 INPUT A3
 890 IF INT A3<>A3 THEN GOTO 880
 900 PRINT A3
 910 IF A1+A2+A3<=Q(P) THEN GOTO 970
 920 PRINT AT 21,9;"GENTE DEMAIS"
 930 FOR F=1 TO 22
 940 IF INT A3<>A3 THEN GOTO 890
 950 NEXT F
 960 GOTO 590
 970 IF Y$<>"PRIMAVERA" THEN GOTO 1040
 980 PRINT "QUANTOS SACOS DE TRIGO SERAO     PLANTADOS?";
 990 INPUT D(P)
1000 IF D(P)>C(P) THEN GOTO 990
1010 PRINT D(P)
1020 LET C(P)=C(P)-D(P)
1030 IF D(P)/10>A2 THEN LET D(P)=A2*10
1040 PRINT AT 21,9;"DIGITE NEW LINE."
1050 IF INKEY$="" THEN GOTO 1050
1060 FOR F=1 TO 22
1070 SCROLL
1080 NEXT F
1090 FAST
1100 PRINT AT 0,0;M$
1110 PRINT AT 8,14;"■■■■■■";AT 9,14;"■    ■";AT 10,14;"■ $$ ■";AT 11,14;"■    ■";AT 12,14;"■■■■■■"
1120 PRINT AT 10,27;"L"
1130 LET T(P)=0
1140 LET Z(P)=0
1150 LET X(P)=0
1160 SLOW
1170 IF A1<Q(P)/2.2 THEN GOSUB 1510
1180 IF A3<Q(P)/2.2 THEN GOSUB 1640
1190 IF D(P)<Q(P)*2 THEN GOSUB 1850
1200 LET C(P)=(C(P)/1.2)+D(P)*3
1210 IF C(P)+D(P)<Q(P)*2 THEN GOSUB 1910
1220 IF C(P)+D(P)>Q(P)*2 THEN GOSUB 2140
1230 LET Q(P)=Q(P)*1.2
1240 LET Q(P)=INT (Q(P))
1250 LET C(P)=INT (C(P))
1260 LET M(P)=M(P)*1.09
1270 LET M(P)=INT (M(P))
1280 FOR F=1 TO 22
1290 SCROLL
1300 NEXT F
1310 IF M(P)<0 THEN LET M(P)=0
1320 IF C(P)<0 THEN LET C(P)=0
1330 IF Q(P)<0 THEN LET Q(P)=0
1340 NEXT P
1350 IF Y<3 THEN GOTO 490
1360 PRINT AT 0,0;"AGORA PARA O INVERNO......"
1370 LET W$=""
1380 LET W=0
1390 FOR F=1 TO A
1400 PRINT N$(F)( TO 10);" COM $";M(F);",  "
1410 IF M(F)>W THEN LET W$=N$(F)( TO 10)
1420 IF M(F)>W THEN LET W=M(F)
1430 NEXT F
1440 PRINT AT 17,0;"PARABENS ";W$
1450 PRINT "VOCE E' O VENCEDOR DO JOGO COM"
1460 PRINT "$";W
1470 PRINT AT 20,0;"QUER JOGAR NOVAMENTE? (S/N)"
1480 INPUT R$
1490 IF R$="S" THEN RUN
1500 STOP
1510 REM ****REPRESA****
1520 IF INT (RND*3)+1=1 THEN RETURN
1530 LET K=INT (RND*10)+5
1540 FOR F=3 TO K+3
1550 FOR G=0 TO 21
1560 PRINT AT G,F;"■"
1570 NEXT G
1580 IF K*10>Q(P) THEN LET K=INT (Q(P)/10)
1590 LET Q(P)=Q(P)-K*10
1600 IF K>8 THEN LET M(P)=M(P)-((K-8)*100)
1610 LET Z(P)=K*10
1620 LET C(P)=C(P)-K*15
1630 RETURN
1640 REM ****DEFESA****
1650 IF INT (RND*3)+1=1 THEN RETURN
1660 LET K=INT (RND*10)+5
1670 IF K*6>Q(P) THEN LET K=INT (Q(P)/6)
1680 FOR F=27 TO 16 STEP -1
1690 PRINT AT 10,F;"L "
1700 NEXT F
1710 FOR F=1 TO K
1720 PRINT AT 10,16;"■ "
1730 PRINT AT 10,16;"■ "
1740 PRINT AT 10,16;" ■"
1750 PRINT AT 10,16;" ■"
1760 NEXT F
1770 FOR F=16 TO 27
1780 PRINT AT 10,F;" L"
1790 NEXT F
1800 LET Q(P)=Q(P)-K*6
1810 LET T(P)=K*6
1820 LET C(P)=C(P)-K*25
1830 LET M(P)=M(P)-K*15
1840 RETURN
1850 REM ****FOME****
1860 LET K=ABS (D(P)-INT (Q(P)*1.5))
1870 IF K>Q(P) THEN LET K=INT (Q(P)/2)
1880 LET Q(P)=Q(P)-(INT (K))
1890 LET X(P)=K
1900 RETURN
1910 REM ****COMPRA****
1920 LET K=INT (RND*5)+15
1930 FOR F=1 TO 22
1940 SCROLL
1950 NEXT F
1960 PRINT AT 0,0;"VOCE NAO TEM TRIGO SUFICIENTE P/ALIMENTAR SUA POPULACAO."
1970 PRINT "VOCE DEVE COMPRAR ALGUM..."
1980 PRINT
1990 PRINT "VALOR ATUAL DE COMPRA = ";K
2000 PRINT
2010 PRINT "VOCE TEM $",M(P)
2020 PRINT
2030 PRINT "O MAXIMO QUE VOCE PODE TER SAO "
2040 PRINT INT (M(P)/K);" SACOS"
2050 PRINT "QUANTOS VOCE QUER COMPRAR ? ";
2060 INPUT I
2070 IF I>(M(P)/K) THEN GOTO 2050
2080 PRINT I
2090 PRINT
2100 PRINT "ISSO VAI CUSTAR-LHE $ ";I*K
2110 LET M(P)=M(P)-(K*I)
2120 LET C(P)=C(P)+I
2130 RETURN
2140 REM ****VENDA****
2150 FOR F=1 TO 22
2160 SCROLL
2170 NEXT F
2180 PRINT AT 0,0;"VOCE TEM UM EXCEDENTE DE TRIGO. SE QUISER VENDER ALGUM, ESPECI- FIQUE A QUANTIA."
2190 PRINT "CASO CONTRARIO,DIGITE ""N"""
2200 PRINT "VOCE TEM :"
2210 PRINT INT (C(P));" SACOS"
2220 INPUT A$
2230 IF A$(1)="N" THEN RETURN
2240 LET K=VAL A$
2250 IF K<=C(P) THEN GOTO 2290
2260 PRINT AT 20,10;"DIGITE NEW LINE"
2270 IF INKEY$="" THEN GOTO 2270
2280 GOTO 2140
2290 LET J=INT (RND*5)+5
2300 PRINT K;" SACOS DE TRIGO ,A"
2310 PRINT "$";J;" O SACO,SAIRA' POR "
2320 PRINT "$";K*J
2330 LET M(P)=M(P)+K*J
2340 LET C(P)=C(P)-K
2350 RETURN
2360 FOR F=1 TO A
2370 CLS
2380 PRINT "JOGADOR ";F
2390 PRINT
2400 PRINT "SEU NOME,POR FAVOR ";
2410 INPUT N$(F)
2420 PRINT N$(F)
2430 PRINT
2440 PRINT "SUA IDADE (APROX.)? ";
2450 INPUT U(F)
2460 PRINT U(F)
2470 PRINT
2480 PRINT "SEU SEXO MASC.(M) OU FEM.(F)? ";
2490 INPUT N$(F)(11)
2500 PRINT N$(F)(11)
2510 PRINT
2520 PRINT "O.K.? (S/N)";
2530 INPUT H$
2540 PRINT H$
2550 IF H$="NAO" OR H$="N" THEN GOTO 2370
2560 NEXT F
2570 RETURN
2580 FAST
2590 LET M$=""
2600 FOR F=1 TO 22
2610 LET M$=M$+"■■■■■■//////"
2620 NEXT F
2630 SLOW
2640 RETURN
2650 SAVE "FEUD■"
2660 RUN
```

a) Os moradores de um feudo podiam ser divididos em servos e vassalos. Os servos estavam encarregados trabalhos de *agricultura* e manutenção do feudo, enquanto que os vassalos eram encarregados da administração e da *defesa* da propriedade. O leitor deve ter em mente que trata-se de pessoas diferentes.

b) Existia dinheiro na Idade Média, mas não era a base do sistema econômico. Neste período, em sua maior parte, o comércio era feito por meio de *trocas de mercadoria.*

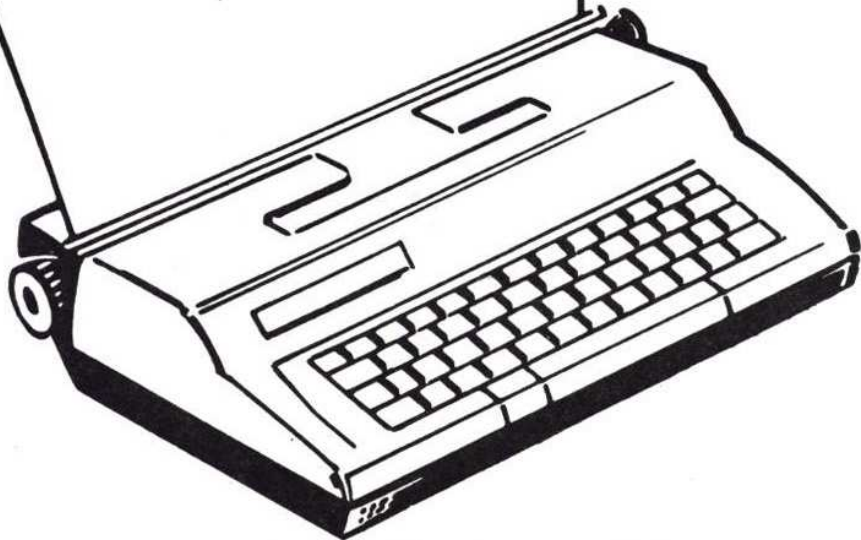
c) Existiam também os ladrões, mas eles não roubavam dinheiro e sim mercadorias, atacando principalmente os mercadores, nas estradas, raramente invadindo um feudo. Os ataques eram provenientes, em sua maioria, de hordas de bárbaros e guerras "particulares" entre senhores feudais.

O FEUDO

INVERNO ANO 1
BARAO ROBERTO
5 PESSOAS VIERAM AO FEUDO.
INFORTUNIOS:
FOME . ENCHENTES . LADROES
0 0 0
****
VOCE TEM:
$56021,00
183 PESSOAS,
1697 SACOS DE TRIGO.
****
TRABALHAM:53 PESSOAS
(A) REPARO DA REPRESA 100
(B) PLANTAR TRIGO 30
(C) DEFESA DO FEUDO 3
DIGITE NEW LINE.
AGORA PARA O INVERNO.....
ROBERTO COM $156254;
PARABENS ROBERTO
VOCE E' O VENCEDOR DO JOGO COM
$156254
QUER JOGAR NOVAMENTE? (S/N)

UM EDITOR DE CARTAS

2ª PARTE

DO TRS—80 PARA O TK

Igor Sartori

A análise do programa desenvolvido em BASIC do TRS-80, que mostramos no número anterior, nos permitirá adaptá-lo ao TK, desde que levemos em consideração, as limitações deste computador.

Para traduzir um programa de um dos dialetos do BASIC, ou mesmo de outras linguagens para o BASIC do TK, nós temos três opções:

a) analisar o programa, instrução por instrução e tentar achar sua equivalente, no dialeto do TK.

b) analisar o programa visando descobrir seu algoritmo e elaborar um programa que execute as mesmas funções.

c) procurar as instruções equivalentes de uma forma genérica e analisar o programa trecho por trecho, visando tanto descobrir o algoritmo, como buscar traduzir linha por linha.

Usaremos o terceiro método por nos permitir, ao mesmo tempo, analisar o algoritmo e as instruções.

SIMPLIFICANDO O PROGRAMA

A nossa primeira preocupação será simplificar o programa, deixando de lado, ao menos por enquanto, a parte que o torna específico para correspondência.

Assim sendo, deveremos nos preocupar apenas com a introdução do texto e a sua correção.

Devido as limitações inerentes ao TK, as linhas deverão ter no máximo 23 caracteres. Para um melhor aproveitamento do papel, sugerimos não usar margens. Desta forma, eliminaremos a ***variável N*** do programa, deixando o limite sempre em 23.

Outra simplificação que faremos é, pelo menos numa fase inicial, eliminar a colocação de algumas mensagens na tela e os comentários (instruções REM). Isso permitirá um melhor aproveitamento da memória. Elas poderão ser colocadas depois, caso ache necessário e disponha de memória suficiente (computadores como o TK podem apresentar memórias de 1k, 2k, 16k, 48k e 56k). Dependendo da memória, apenas uma parte deste programa poderá ser aproveitada. Suporemos, para efeitos práticos, que o computador a ser utilizado tenha uma memória de 16k. Memórias menores exigirão maiores simplificações; memórias maiores permitirão refinamentos maiores, que deixaremos ao cargo do leitor.

AS PEQUENAS ADAPTAÇÕES

Diversas instruções permitem adaptação quase que imediata para o TK. São instruções que têm funções equivalentes ao TK, mas com sintaxe diferente.

Logo na linha 10 encontramos a primeira:

```
10 ' EDITOR DE CARTAS
```

O símbolo **'** indica tratar-se de um comentário e corresponde à REM no TK. Um pouco mais adiante, na linha 20, encontramos:

```
20 CLEAR 2000
```

Em alguns computadores como, por exemplo, o TRS-80, é necessário reservar um espaço na memória para as ***strings***. Isso é feito por meio da instrução CLEAR ***n***, onde ***n*** representa o número de bytes que desejamos reservar. No caso do TK, isso não é necessário, podendo esta linha ser eliminada sem problemas. A seguir, aparecem várias instruções do tipo:

```
30 INPUT "Mensagem", variável
```

No TK, isso não é permitido, mas a solução é simples. Podemos transformar estas linhas em duas, da seguinte maneira:

```
25 PRINT "MENSAGEM"
30 INPUT variável
35 PRINT variável
```

A razão pela qual usamos a linha 35 é que, no BASIC do TRS-80, uma vez digitado o valor pedido pelo INPUT, ele é imediatamente mostrado na tela, o que não ocorre no TK.

Em várias partes do programa, aparece a instrução:

```
PRINT CHR$ (31)
```

A instrução PRINT CHR$ (código) mostra na tela o caracter cujo código está entre parênteses ou executa um comando. No caso do TRS-80, PRINT CHR$(31) significa "limpeza de tela". No TK a instrução que limpa a tela é **CLS.**

Várias vezes encontramos variáveis sendo atribuídas desta forma:

```
190 Z = X
```

Em alguns computadores, a instrução LET pode ser omitida, não ocorrendo o

mesmo com o TK. Devemos, então, colocá-la em todos os pontos onde foi omitida.

Outro caso de omissão é em instrução do tipo:

100 IF condição THEN GOTO número de linha

Aí podem ocorrer dois tipos de omissão: da conjunção THEN ou da instrução GOTO. Ambos os casos não são permitidos no TK.

Outra coisa a ser levada em consideração é o uso, freqüentemente na maioria dos computadores, de mais de uma instrução por linha. Por exemplo:

100 LET A = 0.5: LET B = 1: PRINT A, B

Nesta linha temos três instruções, separadas entre si por dois pontos (:). Na maioria dos casos, podemos traduzir este tipo de linha, dividindo-a em várias linhas, como a seguir:

```
100 LET A = 0.5
101 LET B = 1
102 PRINT A, B
```

Entretanto, em alguns casos, isto não é tão simples. Observe a linha abaixo:

100 IF A$ = "SIM" THEN PRINT "MENSAGEM": GOTO 50

Neste caso, dividir a linha em duas pode ocasionar erro. Vamos analisar o que faz esta linha: se a variável A$ contiver a palavra "SIM", então deve-se imprimir a mensagem e ir para a linha 50. Senão, o programa seguirá para a linha seguinte. Se desdobrarmos esta linha em dois, sem muita atenção, teremos:

```
100 IF A$ = "SIM" THEN PRINT "MENSAGEM"
101 GOTO 50
```

Isso está errado, porque só queremos que o programa siga para a linha 50, se e somente se A$ for igual a "SIM". No exemplo acima, fizemos exatamente o contrário! Neste caso em particular, podemos adotar uma das seguintes soluções:

a) repetindo a condicional

101 IF A$ = "SIM" THEN GOTO 50

b) alterando a condição

```
100 IF A$ <> "SIM" THEN GOTO 110
101 PRINT "MENSAGEM"
102 GOTO 50
110 . . . sequência do programa . . .
```

c) verificar se em outro trecho de programa podem ser feitas alterações. Por exemplo, na linha 50:

50 PRINT "MENSAGEM"
51 . . . instrução que anteriormente estava na linha 50 . . .
. . .
100 IF A$ = "SIM" THEN GOTO 50

Evidentemente, devemos verificar se o programa comporta estas alterações. Cada caso é um caso. . .

STRINGS

Operações com *strings* são bastante comuns em BASIC e, por esse motivo, existe um grupo de instruções destinadas a manipular este tipo de dado.

E o que são *strings*? Podemos considerar uma *string* como um conjunto de caracteres, formado por letras números e outros sinais gráficos. Por exemplo:

123 É UM VALOR NUMÉRICO

No nosso programa, várias funções e instruções que tratam com este tipo de dado foram usadas. Descreveremos cada uma delas, bem como outras que aparecem em computadores da "família" TRS-80, comparando com instruções semelhantes do TK 82/83/85.

VAL (string) — esta função fornece o valor numérico de uma *string* formada por caracteres numéricos. Os caracteres não numéricos que por ventura aparecem após o valor numérico, são ignorados. No TK esta função funciona de modo diferente: ela fornece o resultado de uma expressão numérica que tenha sido, por algum motivo, escrita como *string*. Por exemplo, a linha imediata.

PRINT VAL("SQR(4)+10")

fornecerá o valor numérico 12. Num computador da família TRS-80 forneceria o valor zero!

Esta característica torna o TK, *neste aspecto,* mais poderoso que computadores "parentes" do TRS-80. Entretanto, recomendamos cuidado ao traduzir programas que contenham esta instrução.

LEN(string) — esta função fornece o comprimento de uma *string,* ou seja, o número de caracteres que formam esta *string.* A sintaxe e execução da função é idêntica no TK.

ASC(string) — esta função fornece o código do primeiro caracter, de acordo com o código padrão ASCII. No TK-82 temos a função CODE, que fornece o código do primeiro caractere de uma *string,* segundo um código próprio do TK. Cuidado ao traduzir esta instrução!

CHR$(valor numérico) — fornece o caracter especificado pelo código entre parênteses, quando combinada com a instrução PRINT, ou executa a função especificada, segundo um código próprio do computador (veja a instrução PRINT CHR$(31), déscrita num dos itens anteriores. No TK encontramos uma instrução semelhante que não executa a função indicada pelo código, mas imprime o grupo de caracteres que forma seu mnemônico. Por exemplo:

PRINT CHR$(238),

mostrará na tela do TK:

INPUT

Como na instrução **ASC,** o código que aparece entre parênteses no TRS-80, e na maioria dos computadores, é o código **ASCII** e no **TK** é seu código particular. (Cuidado, portanto).

STR$(número ou expressão numérica) — é o contrário da instrução VAL, pois transforma um valor numérico em uma *string.* O TK possui esta função, com sintaxe idêntica.

INKEY$ — fornece o caractere digitado no teclado. No TK, existe uma função idêntica.

TIME$ — fornece um relógio, com horas, minutos e segundos. Não existe equivalente no **TK** e sua "tradução" exige um programa muito grande, não compensando sua implementação, devido ao grande espaço ocupado na memória.

STRING$(n, "caracter") — esta função fornece uma sequência de *n* caracteres idênticos. Por exemplo:

PRINT STRING$(10, "*")

mostrará na tela:

****

Não existe esta função no TK. Ela poderá ser substituída, em alguns casos, por:

10 FOR I = 1 TO N (N é o número de repetições desejado)
20 PRINT "*";
30 NEXT I

Em outros casos, devemos analisar o programa para tornar possível a sua tradução.

Existem, ainda, instruções destinadas ao fatiamento de *strings,* que discutiremos na próxima parte.

Tanios Hamzo.

MONI TK
Transforme a sua TV num monitor de vídeo

Uma das mais marcantes diferenças entre os computadores da família do SINCLAIR ZX81 (TKs, NE-Zs, CPs, . . .) e aquelas de maior porte físico, é sem dúvida, o monitor de vídeo.

Indiscutivelmente melhor, a imagem de um monitor de vídeo poderá ser conseguida em seu próprio aparelho de TV, com pouco custo.

Para quem não sabe, numa configuração convencional o micro "finge" que é um canal de TV (geralmente o canal 2) e envia seus sinais para o aparelho de TV, que por sua vez pensa que está recebendo os sinais de uma estação de TV. A figura 1 mostra o que acontece com estes sinais desde a sua criação até a sua aparição na tela de TV. Note que o caminho é longo, porque tem que se modular o sinal que será demodulado pela TV. Um paradoxo.

No sistema modificado (que chamamos MONI-TK), este caminho é drasticamente reduzido, "cortando o caminho" convencional. A figura 2 mostra como o MONI-TK funciona, pegando um atalho do controle de vídeo do micro até o controle de saída de vídeo da TV, direto e sem escalas. Com isto, acaba-se tendo muitas vantagens:

Melhor imagem, livre de interferência externa e distorções, já que os sinais vão "puros" do micro à TV, eliminando assim os incomodos ajustes finos de sintonia e operando em qualquer canal VHF ou UHF, pois não passam pelo seletor de canais (você pode inclusive ouvir um programa enquanto opera o micro).

Reversibilidade ao sistema convencional a qualquer tempo, simplesmente mudando a posição da chave (você pode dar uma olhadinha na TV para ver se a novela já começou enquanto o micro calcula aquela integral cabeluda).

Maior contraste, e com isso menor "cansaço visual". Aquele caracter gráfico xadrezinho (CHR$ 8) sairá como no desenho do manual.

Aliados a tudo isto, somam-se a simplicidade das modificações que não exigem profundos conhecimentos de eletrônica prática e o pequeno número de componentes (apenas 4), considerados baratos.

Se você não possui conhecimento técnico suficiente para a empreitada (dentro da TV há tensões da ordem de 20.000 volts), procure um técnico em TV de confiança.

Modificações no micro

Tomaremos como exemplo o TK82-C, porém, a princípio, poderia ser qualquer micro que use uma TV como forma de visualizar sua operação.

Primeiramente abra o gabinete do micro, soltando seus 5 rebites plásticos e removendo a parte superior. Tome cuidado com a fiação do joystick. Localize o modulador de vídeo e os dois fios a ele ligados (fig. 3); corte o fio de sinal de vídeo em sua parte média e solde uma extensão de fio com aproximadamente 10 cm em cada extremidade deste fio cortado.

Você deve proteger o circuito de eventuais respingos de solda fundida, cobrindo-o com uma folha de papel.

Em um plugue "JACK" fêmea com interrupção (aqueles com 3 terminais), similar aos do micro, solde um fio com comprimento de uns 10 cm entre seu ponto de terra e o do micro (um bom lugar é a carcaça do modulador). Conforme esquematizado na figura 4, solde o plugue aos outros fios, certificando-se que o fio que está ligado ao modulador não entrará em contato com o plugue macho quan-

fig. 1

MICROCOMPUTADOR

APARELHO DE TELEVISÃO

SISTEMA MODIFICADO

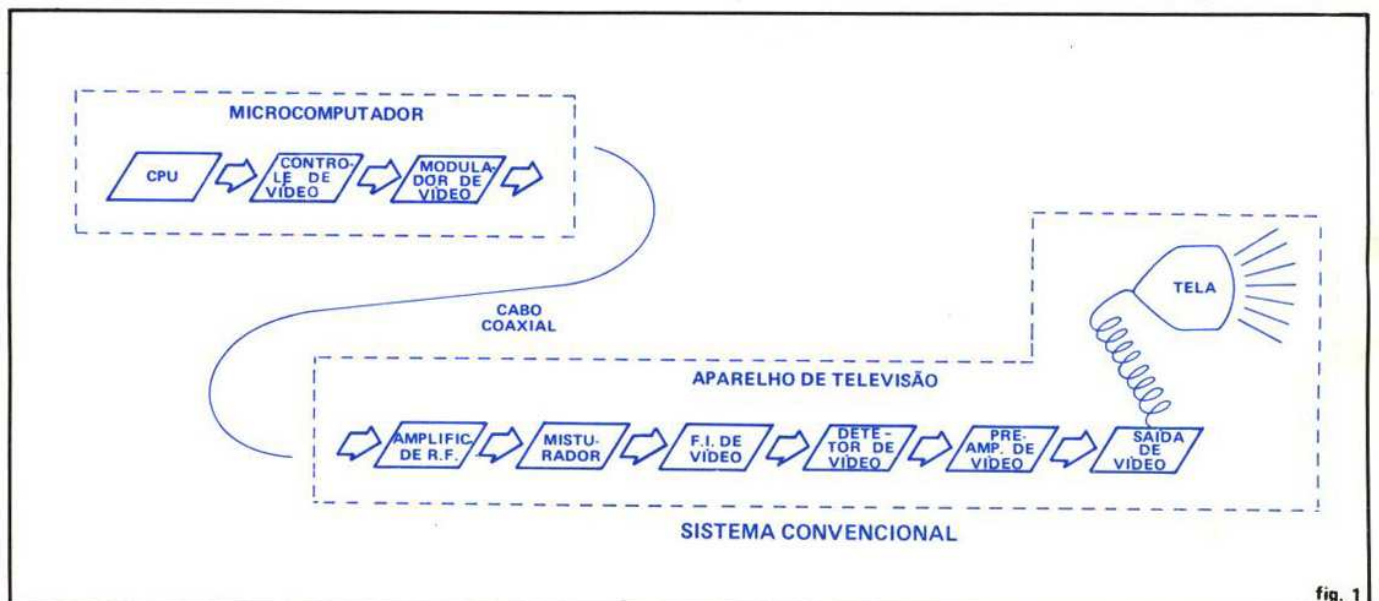
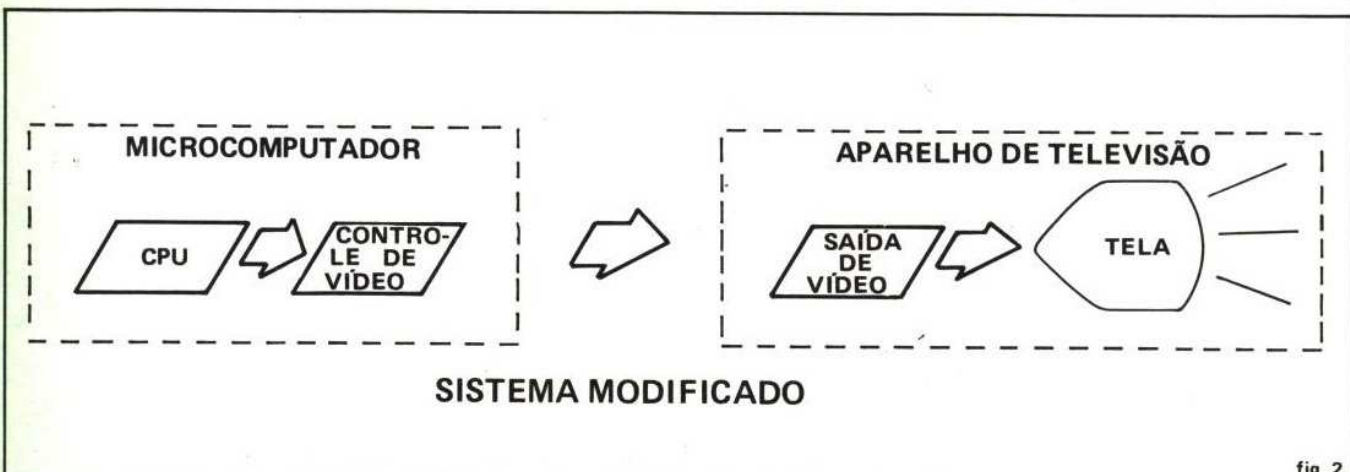



fig. 2

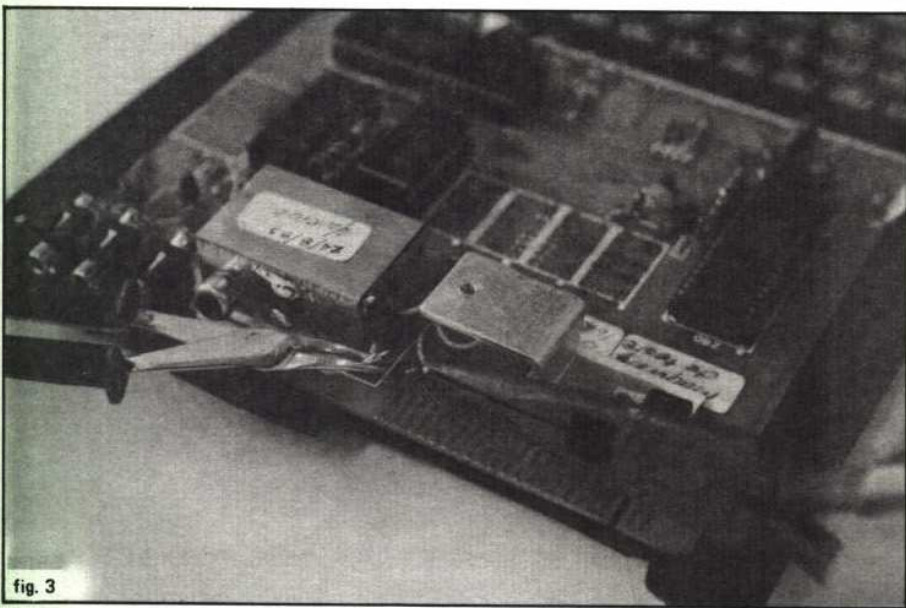



fig. 3

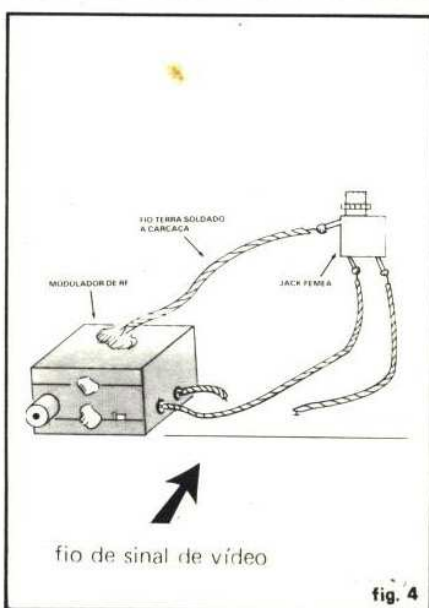



fig. 4

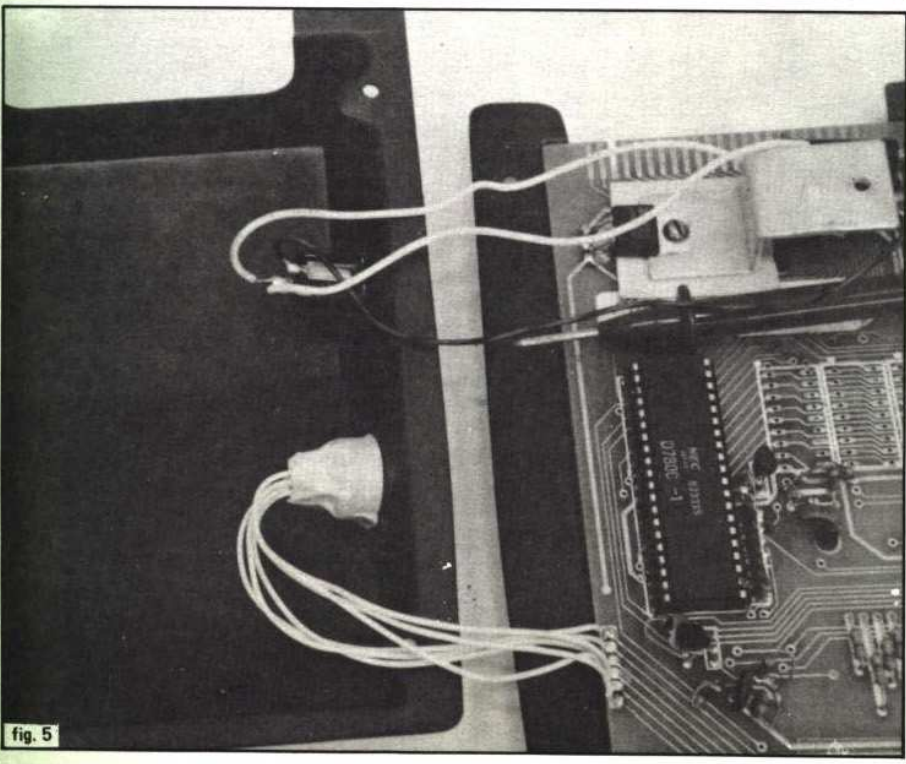



fig. 5

do este for conectado. Isole todas as soldagens.

Com um canivete ou estilete, abra um furo na tampa do computador, numa posição cômoda (um bom lugar é sua lateral esquerda) e prenda aí o plugue (fig. 5), recoloque a tampa e os rebites.

Ligue o micro da maneira convencional; a imagem deverá aparecer como de costume e sumir quando um jack macho for ligado à nova saída (que chamaremos de "saída de sinal de vídeo") porque se usada, desligará a saída convencional.

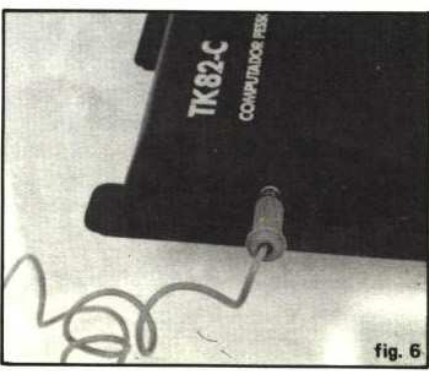



fig. 6

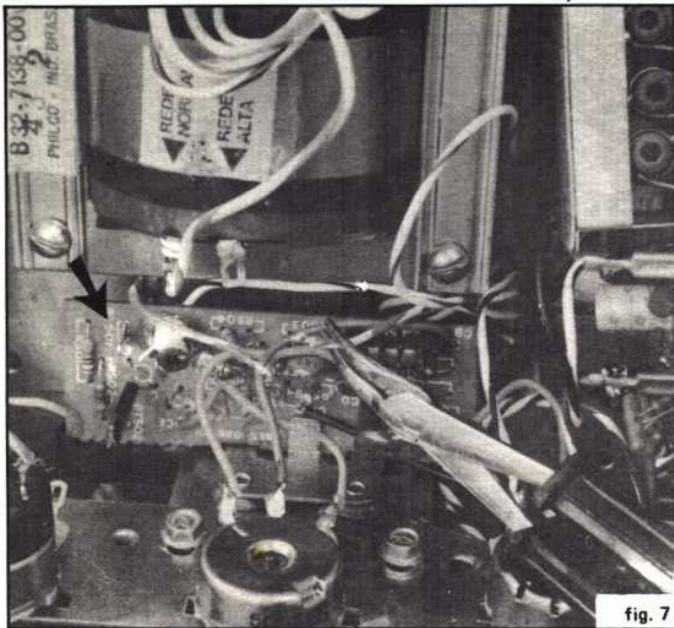



fig. 7

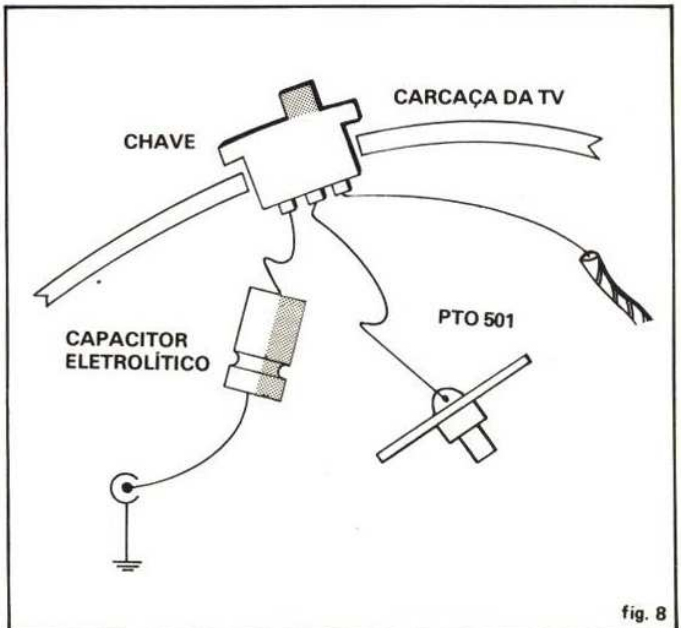



fig. 8

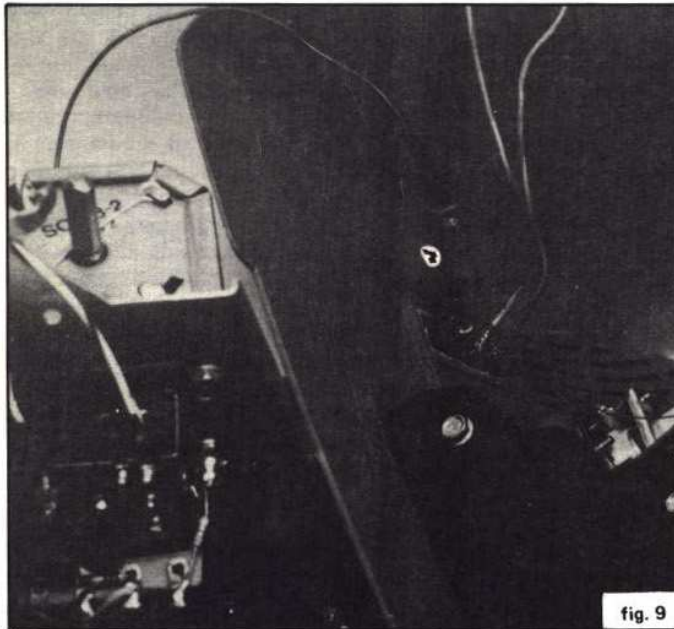



fig. 9

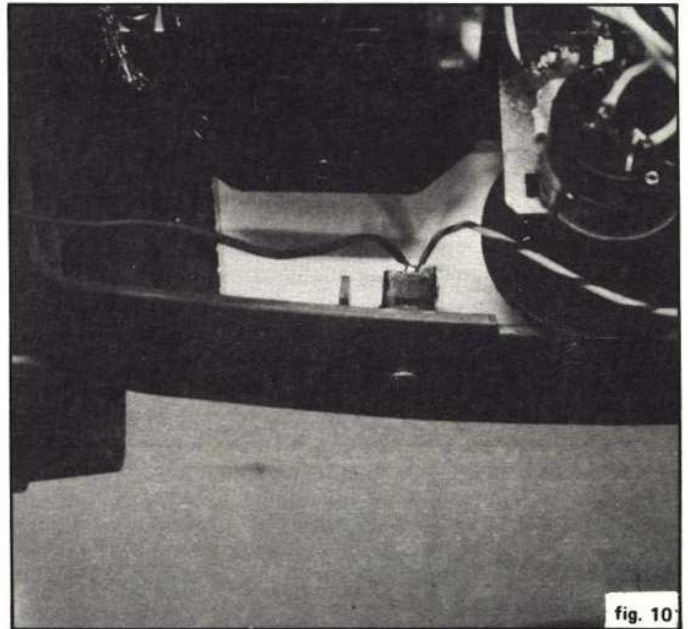



fig. 10

Modificações no aparelho de TV

Teoricamente, qualquer aparelho de TV pode ser adaptado. Como exemplo, escolhemos um Philco modêlo B-265-2M, que é bastante popular.

Primeiramente desligue o aparelho da tomada; depois, sobre uma almofada ou travesseiro, "deite" a TV com a "cara" (tela) para baixo.

Munido de uma chave de bôca de 1/4 de polegada, solte os 5 parafusos que prendem a tampa traseira da TV. Com a antena para fora do aparelho e desconectada, abra a TV e localize, próxima ao seletor de canais uma placa de circuito impresso ao lado dos botões de controle (volume, etc.). Nesta placa, há um pino com um código gravado perto: PT 501; o que significa "ponto 501". Deste ponto, do outro lado da placa, está soldado um fio, que em nosso caso era listrado de branco e verde (figura 7).

Puxe para si este fio que sai do ponto 501 com cuidado e corte-o a cerca de 5 cm da placa. À cada extremidade, solde um fio de extensão próxima a 30 cm.

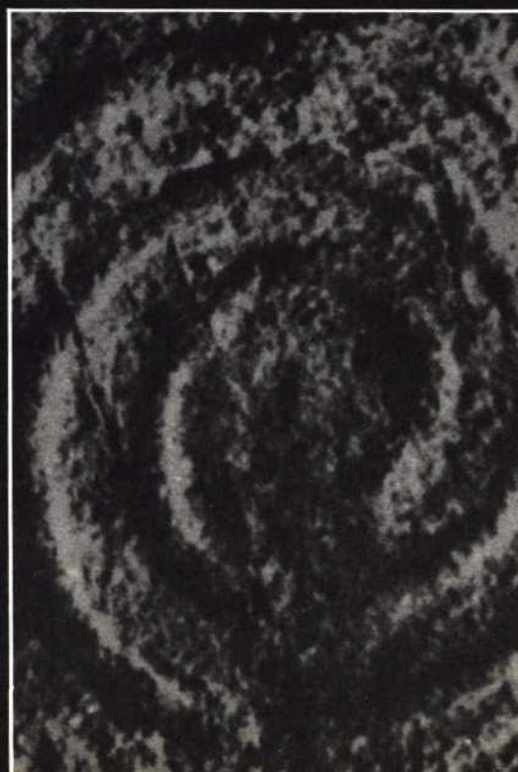
Acompanhe pela figura 8 as próximas ligações. Ao pino central de uma chave de um pólo e duas posições, ligue o fio que tem a outra extremidade ligada à placa. A um dos pinos laterais da chave, solde o outro fio.

Ao terminal restante da chave, solde o pólo negativo de um capacitor eletrolítico de 100 microfarads por 40 volts (ou mais). Quanto ao pólo positivo, solde-o a um novo fio, com cerca de 20 cm; à sua outra extremidade, solde o pino central (o que entra em contato com o plugue macho) de um outro plugue jack fêmea.

Ligue o terminal de terra deste plugue, através de um fio de comprimento adequado à carcaça metálica da TV.

Escolha locais apropriados para instalar a chave MONITOR/TV e o plugue fêmea. Finalmente isole todas as ligações, principalmente as do capacitor; monte a TV e teste seu funcionamento como TV, como monitor de vídeo e como receptor de sinais do micro (sistema convencional).

Se tudo correu bem, você possui uma TV/MONITOR DE VÍDEO de fazer inveja a muito monitor importado por aí.

Aguardem para daqui a alguns números, um outro circuito para incrementar sua TV.

VERMES

Renato da Silva Oliveira

Os primeiros programas *vermes* surgiram à cerca de duas décadas atrás, quando paleontólogos procuravam entender o significado de alguns sulcos nas superfícies de rochas formadas a partir da fossilização de lamas antigas. Tais sulcos, provavelmente, são traçados fósseis de vermes pré-históricos e, devido ao fato de formarem figuras surpreendentemente ordenadas, intrigaram os paleontólogos ao ponto de levá-los a simular, em computadores, os supostos movimentos dos vermes.

Dessa forma, surgiram os programas que desenhavam trajetórias como as mostradas na figura 1. Algum tempo depois do surgimento desses primeiros ***programas vermes***, Seymour A. Pappert, trabalhando

→

no Artificial Intelligence Laboratory do MIT construiu uma série de animais robôs, que se moviam controlados por um pequeno computador programado com instruções simples.

O mais conhecido dos *animais* de Pappert é sua ***Tartaruga.*** A Tartaruga era posta a caminhar sobre uma superfície extensa coberta por papel branco com uma caneta presa ao seu lado, para registrar, no papel; sua trajetória. Figuras geométricas belíssimas eram geradas através de um pequeno número de instruções e, devido à semelhança com as simulações dos vermes pré-histórico, designou-se a todos esses programas, indistintamente, de *programas vermes.*

As instruções dos ***vermes*** de Pappert podem ser utilizadas em qualquer computador que disponha de tela ou impressora, sendo que o ***animal robô*** deve ser substituído por um ponto. Nesse caso, pode-se utilizar a instrução PLOT para traçar a trajetória do ***verme*** (ponto). Uma sequência de instruções típica dos ***programas vermes*** para computadores com tela é por exemplo, a seguinte:

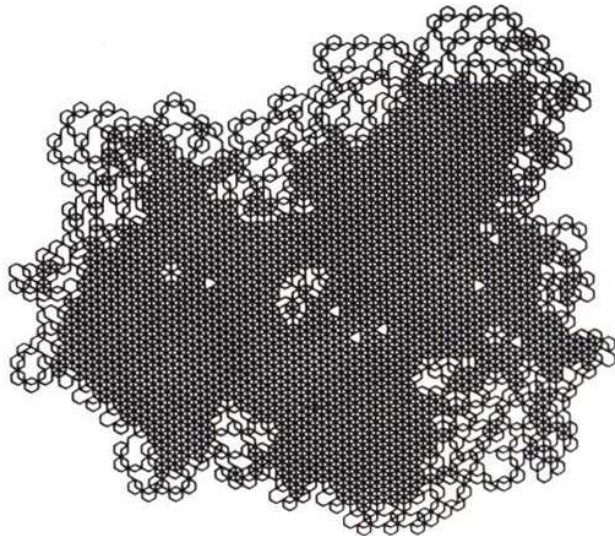
1. Parta do ponto de coordenada (X,Y)
2. Ande F unidades para frente
3. Ande C unidades para cima
4. Ande T unidades para traz
5. Ande B unidades para baixo
6. Repita tudo a partir do procedimento 2

Uma outra sequência de instruções simples, mas de surpreendente versatilidade é a que se segue:

1. Parta do ponto de coordenadas (H,V)
2. Ande P unidades em frente
3. Gire A graus em relação ao último trecho percorrido
4. Ande P+D unidades nessa nova direção
5. Tome P+D para novo valor de P
6. Repita os procedimentos 3, 4 e 5 T vezes
7. Tome o valor inicial de P como seu novo valor
8. Repita tudo a partir do procedimento 2

A diversidade dos padrões gerados por essa seqüência pode ser verificada através do programa abaixo. Introduza-o (mantendo a numeração das linhas) e rode-o:

FIGURA 2

```
 30 INPUT H
 50 INPUT V
 70 INPUT P
 90 INPUT A
110 INPUT D
130 INPUT T
230 LET I=H
240 LET J=V
250 LET G=0
260 LET Q=P
270 FOR N=1 TO T
280 FOR R=0 TO Q-1
310 LET X=INT (I+0.5+R*COS G)
360 LET Y=INT (J+0.5+R*SIN G)
410 PLOT X,Y
420 NEXT R
430 LET I=INT (I+0.5+R*COS G)
440 LET J=INT (J+0.5+R*SIN G)
460 LET Q=Q+D
500 LET G=G+A*PI/180
530 NEXT N
540 GOTO 260
```

Você deverá introduzir seis números no TK. O primeiro e o segundo são respectivamente a coordenada horizontal e vertical do ponto de partida. Para introduzí-los, lembre-se que a tela é dividida em 64 linhas horizontais e 44 verticais.

O terceiro, é o número de pontos a ser plotado no primeiro trecho.

O quarto é o ângulo de giro em relação ao último trecho percorrido, dado em graus e respeitando a convenção de sentidos da circunferência trigonométrica.

O quinto número é a diferença entre os ***comprimentos*** de dois trechos consecutivos.

O sexto e último número é a quantidade de trechos a serem traçados.

Para testar o programa, introduza os valores relacionados abaixo:

	Ordem	Valor a ser digitado
Coordenadas do ponto de partida	1º 2º	32 22
Pontos no 1º trecho	3º	2
Ângulos de Giro	4º	90
Diferença entre dois trechos consecutivos	5º	1
Número de trechos	6º	9

O padrão desenhado na tela deve ser semelhante ao da figura 3.

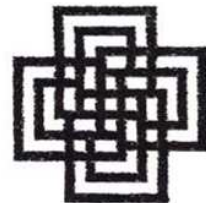
FIGURA 3

Se você testar algumas outras seqüências, notará que, apesar de versátil, esse programa possui algumas deficiências. Por exemplo, quando o ***verme*** parece querer sair da tela, ou ele é ***refletido*** ou o programa é interrompido com denotação de erro ***B.*** Isso pode ser corrigido introduzindo-se as seguintes linhas:

FIGURA 4

```
320 IF X<0 THEN LET X=X+64
330 IF X<0 THEN GOTO 320
340 IF X>63 THEN LET X=X-64
350 IF X>63 THEN GOTO 340
370 IF Y<0 THEN LET Y=Y+44
380 IF Y<0 THEN GOTO 370
390 IF Y>43 THEN LET Y=Y-44
400 IF Y>43 THEN GOTO 390
```

Essas linhas tornarão a tela ***esférica,*** no sentido de que sua superfície fica ilimitada e o ***verme*** pode ***enrolar-se*** nela.

Outro aspecto negativo do programa é a morosidade do ***verme.*** Pode-se evitá-la, acrescentando as linhas a seguir:

FIGURA 5

```
290 IF INKEY$="F" THEN FAST
300 IF INKEY$="D" THEN SLOW
```

Elas possibilitam a escolha da velocidade enquanto o programa é lido. Digite a seqüência abaixo para testar o programa com essas dez linhas a mais. Durante a execução, pressione as teclas ***F*** e ***D*** alternadamente e observe que o modo de operação muda de SLOW para FAST ou de FAST para SLOW.

	Ordem	Valor a ser digitado
Coordenadas do ponto de partida	1º 2º	60 40
Pontos no 1º trecho	3º	4
Ângulos de Giro	4º	90
Diferença entre dois trechos consecutivos	5º	2
Número de trechos	6º	4

Variável	Ordem	1ª sequência Valor a ser digitado	2ª sequência Valor a ser digitado	3ª sequência Valor a ser digitado	4ª sequência Valor a ser digitado
H	1ª	26	26	26	26
V	2ª	16	2∅	16	16
P	3ª	21	21	9	3
A	4ª	135	135	−9∅	12∅
D	5ª	−7	−7	−3	3
T	6ª	3	3	5	5
O$	7ª	N = 5	(N = 1 OR N = 2)	(N = 3 OR N = 4)	N = 8
O	8ª	5	3	6	8
M$	9ª	N = 3	N = 5	N = 2 OR N = 3 OR N = 4	N = 2
M	10ª	−135	∅	9∅	−12∅

Apesar de, com esse programa, o número de variações ser bastante grande, podemos pensar em introduzir algumas informações a mais para o *verme*, afim de que ele mude o ângulo de giro (A) e/ou a diferença entre dois trechos consecutivos (D), durante a execução do programa.

Dessa forma, reescrevemos a instrução *6*, tornando-a pouco mais complexa:

a. Repita os procedimentos 3, 4 e 5 *T* vezes.

b. Quando for executar a *O$-ésima* vez, faça a diferença entre dois trechos consecutivos igual a *O*.

c. Quando for executar a *M$-ésima* vez, faça o ângulo de giro igual a *M*.

Traduzindo isso para a linguagem Basic do TK, obtemos as linhas a serem acrescentadas ao nosso programa:

FIGURA 6

```
150 INPUT O$
170 INPUT O
190 INPUT M$
210 INPUT M
450 IF VAL O$ THEN GOTO 480
470 GOTO 490
480 LET Q=Q+O
490 IF VAL M$ THEN GOTO 520
510 GOTO 530
520 LET G=G+M*PI/180
```

Observe que, sendo *N* a variável que controla quantas das *T* vezes já foram executadas, as "STRING's" *O$* e *M$* devem ser do tipo:

"N= "
"N> "
"N< "
"N>="
"N<="
"N<>"
"N=... OR N= ... OR N= ...
... etc ..."

.
.
.

"etc..."

Note que elas são constituídas basicamente de instruções lógicas. Isso é o que aumenta, em muito, seu controle sobre o *verme.*

Agora você terá que introduzir oito números e duas condições no micro, estas últimas correspondendo às STRING's *O$* e *M$*. Tente as sequências a seguir e compare seus resultados:

Você pode ainda acrescentar ao programa, algumas linhas para torná-lo mais compreensível durante a execução. A seguir, encontra-se listada uma sugestão de programa completo:

FIGURA 7

```
 10 REM "VERMES"
 20 PRINT AT 1,1;"H=";AT 2,1;"V
=";AT 3,1;"P=";AT 4,1;"A=";AT 5,
1;"D=";AT 6,1;"T=";AT 7,1;"O$:";
AT 8,1;"O=";AT 9,1;"M$:";AT 10,1
;"M="
 30 INPUT H
 40 PRINT AT 1,3;H
 50 INPUT V
 60 PRINT AT 2,3;V
 70 INPUT P
 80 PRINT AT 3,3;P
 90 INPUT A
100 PRINT AT 4,3;A
110 INPUT D
120 PRINT AT 5,3;D
130 INPUT T
140 PRINT AT 6,3;T
150 INPUT O$
160 PRINT AT 7,4;O$
170 INPUT O
180 PRINT AT 8,3;O
190 INPUT M$
200 PRINT AT 9,4;M$
210 INPUT M
220 PRINT AT 10,3;M
224 PAUSE 60
225 CLS
230 LET I=H
240 LET J=V
250 LET G=0
260 LET Q=P
270 FOR N=1 TO T
280 FOR R=0 TO Q-1
290 IF INKEY$="F" THEN FAST
295 IF INKEY$="Z" THEN COPY
300 IF INKEY$="D" THEN SLOW
305 IF INKEY$="V" THEN CLS
310 LET X=INT (I+0.5+R*COS G)
320 IF X<0 THEN LET X=X+64
330 IF X<0 THEN GOTO 320
340 IF X>63 THEN LET X=X-64
350 IF X>63 THEN GOTO 340
360 LET Y=INT (J+0.5+R*SIN G)
370 IF Y<0 THEN LET Y=Y+44
380 IF Y<0 THEN GOTO 370
390 IF Y>43 THEN LET Y=Y-44
400 IF Y>43 THEN GOTO 390
410 PLOT X,Y
420 NEXT R
430 LET I=INT (I+0.5+R*COS G)
440 LET J=INT (J+0.5+R*SIN G)
450 IF VAL O$ THEN GOTO 480
460 LET Q=Q+D
470 GOTO 490
480 LET Q=Q+O
490 IF VAL M$ THEN GOTO 520
500 LET G=G+A*PI/180
510 GOTO 530
520 LET G=G+M*PI/180
530 NEXT N
540 GOTO 260
```

A linha 295 permite tirar uma cópia da tela, na impressora, mediante a digitação da tecla *Z* e a linha 3∅5 permite apagar a tela digitando a tecla *V*.

Ao contrário do que pode parecer à primeira vista, as utilidades desses programas são inúmeras. Basta dizer que Pappert (que já foi assistente de Jean Piaget) utilizou seus programas para ensinar matemática às crianças, particularmente geometria. A idéia, nesse caso, é conceituar figuras geométricas como "caminhos percorridos por vermes". Um quadrado, por exemplo, é gerado pelas seguintes instruções:

1. Parta do ponto (I, J)
2. Ande *P* unidades
3. Gire 90 graus
4. Repita tudo a partir do procedimento 2.

Outro importante fato relacionado com os programas *vermes* são suas contribuições ao desenvolvimento da linguagem LOGO, tida por muitos *experts* em programação, como sucessora da BASIC e muito superior a ela.

O leitor também deve ter percebido que é extremamente difícil prever a maioria dos padrões traçados pelos *vermes,* apesar de eles seguirem umas poucas e simples instruções. Hoje, desenvolvem-se novas áreas da matemática, através de teorias criadas exclusivamente com intuito de tentar prever o padrão produzido por um verme, a partir de suas instruções. Talvez seja esse o aspecto mais importante a ser abordado nos programas *vermes.* Essas teorias, uma vez desenvolvidas, poderiam ser utilizadas em diversas outras áreas do conhecimento humano.

A partir dos programas de Pappert, surgiram outros tipos de programas *vermes.* Esses programas, cujas origens ligam-se aos nomes de John H. Conway, Michael Paterson e Michael Beeler, são ainda mais surpreendentes e talvez os abordemos numa próxima oportunidade.

Não só pela beleza dos padrões gerados (utilizáveis até mesmo por artistas), mas por sua interligação fecunda com outras áreas do conhecimento, os programas *vermes* nos mostram quão simpáticos e úteis podem ser esses animais, desde que saibamos como olhá-los.

Os vermes pré-históricos que tinham em seus *programas genéticos* as instruções para locomoverem-se em trajetórias regulares, provavelmente desencadearam a construção de mais um andar do enorme edifício matemático que conhecemos hoje.

○

NOVIDADES

O lançamento do TK-83 e a obsolescência planificada

O TK-82C veio para atender uma faixa de público que desejava um computador simples, confiável e, principalmente, barato.

Este computador provou, ao longo de sua existência, ser o melhor de sua categoria apresentando características que não o deixavam em desvantagem, quando comparado com modelos maiores.

Entretanto, quando a Microdigital lançou o TK-85, alguns leitores ficaram temerosos. Será que não era um caso de "obsolescência planificada"?

Obsolescência planificada?

Isso significa o lançamento de um produto mais sofisticado em relação a um anterior, em curto espaço de tempo, obrigando os possuidores do modelo antigo se desfazerem dele, comprando o novo.

Será que isso vai acontecer conosco, possuidores do TK-82C?

Notamos que, em pouco tempo, foram lançados o TK-82C, o TK-85 e o TK-2000 color. Embora cada um destes produtos atinja uma faixa de público diferente, só isso e as promessas do fabricante, não seriam suficientes para convencer um possuidor do TK-82C da "pureza de intenções" de seus fabricantes. Mas não foi com promessas que a Microdigital respondeu a seus clientes, lançando um novo produto: o TK-83.

Mais um? Não iremos "dançar" com este novo produto?

Certamente que não! O TK-83 é totalmente compatível com o TK-82C, apresentando as mesmas funções e a mesma capacidade de memória, admitindo as mesmas expansões e o mesmo *software*.

Por que, então, lançar um produto novo?

O TK-83 apresenta um novo *design*, mais moderno, dando um visual mais estético e funcional ao computador. Além disso, a nova caixa é mais resistente, protegendo melhor o *hardware* contra choques e esforços mecânicos. Além disso, o circuito foi desenhado de forma a tor-

nar-se mais compacto, com uma melhor distribuição dos componentes, o que facilita a manutenção do computador.

Com o lançamento do TK-83, tanto os possuidores do TK-82C quanto aqueles que desejam um computador desta categoria, podem dispor de um produto confiável sob diversos pontos de vista, inclusive quanto ao apoio de manutenção e ***software.*** Além disso, o novo computador terá um apoio extra da Microhobby, que publicará sempre programas que possam ser rodados em computadores desta categoria, com ou sem expansão de memória.

A Informática presente em palestras durante o mês de novembro

A Servimec realizou, em São Paulo, uma série de seminários abordando temas técnicos na área da informática, destinados a gerentes de sistemas, analistas e programadores.

Inaugurando a programação prevista, a empresa trará Jean Dominique Warnier — considerado a maior autoridade em informática da França — que dará um curso sobre o que mais entende: "A Metodologia Warnier".

Também em São Paulo, a AIESEC — Association Internationale des Etudiants en Sciences Economiques et Commerciales — esteve realizando seminários (de 8 a 11 de novembro às 20 horas), com o tema "Impacto Social da Informática". O evento contou com a presença de diversos profissionais da área, entre eles o nosso diretor Pierluigi Piazzi.

Uma novidade na prestação de serviços aos usuários e adeptos da computação

Inaugurada na zona sul de São Paulo, a loja matriz da Computique Comércio e Exportação de Computadores. Na inauguração, a empresa ofereceu três coquetéis com o objetivo de reunir pequenos grupos entre fornecedores, amigos, clientes e a imprensa; desta forma, possibilitou a estes, o conhecimento da filosofia das lojas, ou seja, a prestação de serviços e a venda de soluções para aqueles que as procuram.

Clube do Micro

É uma iniciativa do CENADIN — Centro Nacional do Desenvolvimento da Informática, prestador de serviços na área da informática e que oferece cursos de micros a estudantes de 1º e 2º graus. O clube objetiva oferecer um ponto de encontro aos ***adeptos*** do uso de micros.

COMO FAZER SUA ASSINATURA

Para obter seu exemplar mensal (12 números) da revista **Microhobby** contendo muitos programas para o TK como também para o Apple e o TRS-80, inúmeras dicas e as últimas novidades na área de informática, você deve fazer uma assinatura.

É importante ressaltar que o recebimento da revista é considerado a partir da data de recebimento do pedido de assinatura, porém, há um período de 30 dias de *carência* até a revista chegar às suas mãos. Os números anteriores da revista podem ser adquiridos se as tivermos no momento do pedido em nossos estoques, através do telefone 256-8348, diretamente com o Departamento Comercial.

Para ter acesso a todas estas vantagens basta preencher, corretamente, o cupom anexo, colocá-lo no correio junto a um cheque nominal ou vale postal em nome de Micromega Publicações e Material Didático, no valor de Cr$ 14.800,00.

O envelope deverá ser selado e endereçado com os seguinte dizeres:

MICROMEGA P.M.D. LTDA.
Departamento de Assinaturas
Caixa Postal 54121
CEP 05096 – São Paulo, SP.

e dentro dele não deverá ter nada além do cheque e o cupom.

No verso do cheque, escreva:

"Destina-se ao pagamento de uma assinatura (12 números) da revista Microhobby".

Quando este cheque for devolvido ao seu Banco com nosso endôsso, servirá (para você), de comprovante provisório até que nosso recibo seja confeccionado e enviado pelo correio.

SORTEIO: RESULTADOS

Resultado do sorteio realizado na III Feira da Informática

dia 17/10/83
1º PRÊMIO: nº 000782 – REGINA GIANOLLA
(sacola com periféricos, fitas e assinatura da Microhobby por 6 meses)
2º PRÊMIO: nº 001693 – RICHARD DRABEK
(placa mãe da TPLAK)

dia 18/10/83
1º PRÊMIO: nº 005894 – MOISES JOEL PLATTEK
(3 fitas)
2º PRÊMIO: nº 002505 – RONALDO PIRES MONTEIRO
(Compilador Tigre)
3º PRÊMIO: nº 002480 – ANTONIO NEGREIRA LOPEZ
(Compilador Tigre)
4º PRÊMIO: nº 002062 – CIRO TADACHI FUZIHARA
(Compilador Tigre)
5º PRÊMIO: nº 001956 – PAULO SOARES OLIVEIRA FILHO
(Placa TPLAK)
6º PRÊMIO: nº 002034 – ROSECLEA LOPES DE OLIVEIRA MELO
(Tig-Loader)
7º PRÊMIO: nº 002459 – HELENA PIRES DE SOUZA LEITE
(Compilador Tigre)
8º PRÊMIO: nº 004657 – WALTER PARTEL
(fita Simulador de Vôo)

dia 19/10/83
1º PRÊMIO: nº 004343 – JOSÉ CARLOS MOITA
(fitas: Invasores, Compilador Tigre, Mansão Maluca, Simulador de Vôo, Tig-Loader)
2º PRÊMIO: nº 007599 – MAURO ROSSIN FILHO
(Placa Mãe da TPLAK, assinatura da Microhobby por 6 meses)
3º PRÊMIO: nº 007484 – PAULO SERGIO DE OLIVEIRA JORDÃO
(fitas: Mansão Maluca e Simulador de Vôo)

dia 20/10/83
1º PRÊMIO: nº 002426 – ERIVALDO ALVES DOS SANTOS
(placa mãe p/periféricos, Tig-Loader, Compilador Tigre, fita Mansão Maluca e Simulador de Vôo)
2º PRÊMIO: nº 067034 – ASCANIO DE PAIVA
(Tig-Loader, Compilador Tigre, fita Simulador de Vôo)

dia 21/10/83
1º PRÊMIO: nº 024776 – MARCIA RITA CARRARA
(2 fitas de jogos para TK, Tig-Loader, Compilador Tigre, fita Simulador de Vôo)
2º PRÊMIO: nº 027900 – GABRIEL MILBRATZ
(placa mãe da TPLAK, Compilador Tigre)

dia 22/10/83
1º PRÊMIO: nº 031535 – JOSÉ ANTONIO DE SOUZA
(bolsa térmica da Exatron, 2 fitas de jogos p/ TK, Tig-Loader, Compilador Tigre, fita Simulador de Vôo)
2º PRÊMIO: nº 030048 – LUIZ GONZAGA DE OLIVEIRA
(Placa mãe da TPLAK, Compilador Tigre)

dia 23/10/83
1º PRÊMIO: nº 045071 – MIGUEL MASSASHI FIJISHIGE
(bolsa térmica p/ microcomputador, fita Compilador Tigre, Placa Big p/ TK, Tig-Loader, assinatura da Microhobby por 6 meses)
2º PRÊMIO: nº 010437 – JORGE CHRISTOS MELLO VERNARDOS
(placa Bip, fita Compilador Tigre, assinatura Microhobby por 6 meses)

Sorteio Geral

nº 012618 – EVERSON ALBANO
(1 micro computador TK 83, 1 fita da Exatron)

Todos os ganhadores no sorteio, deverão retirar seus prêmios na MICROMEGA PUBLICAÇÕES E MATERIAL DIDÁTICO LTDA. à Rua Bahia, nº 1049 – Pacaembu – São Paulo – SP. Os ganhadores que residirem fora da Grande São Paulo receberão seus prêmios em suas casas.

CHEGA DE PROBLEMAS!

APENAS
Cr$ 18.000,00
Preço válido até 30/12/83

TIG-LOADER

D L IN F

Leoni

use Tig Loader.

TIG-LOADER possibilita:

- *a localização do ótimo volume do gravador, facilitando a operação LOAD.*
- *DUPLICAR qualquer programa, mesmo aqueles "fechados".*
- *carregar (LOAD) e DUPLICAR simultaneamente.*
- *gravar (SAVE) em 2 gravadores ao mesmo tempo.*
- *monitorar as operações LOAD, SAVE ou DUPLICAÇÃO através de fone.*
- *filtrar as interferências elétricas de baixa frequência, que são a causa da maioria dos problemas de LOAD/SAVE.*

NÃO CARREGUE ESTES PROGRAMAS!

V. não precisa mais carregar estes programas: eles estão prontos para uso, em CARTUCHO! Conecte o cartucho, ligue o micro e... só! Seis aplicativos à sua disposição, esperando V. comandar. SEM modificar seu micro. E a expansão de memória é usada normalmente! Como lançamento, o TIGRE oferece: hi-speed, renumerador, apagador de linhas em bloco, soma sintática, disponibilidade de memória e rotinas de vídeo em um único cartucho! É ESPETACULAR!!!

ESCREVA PEDINDO INFORMAÇÕES

TIGRE
COM. DE EQUIP. P/COMPUTADORES LTDA.
Rua Correia Galvão, 224 - CEP 01547 - São Paulo

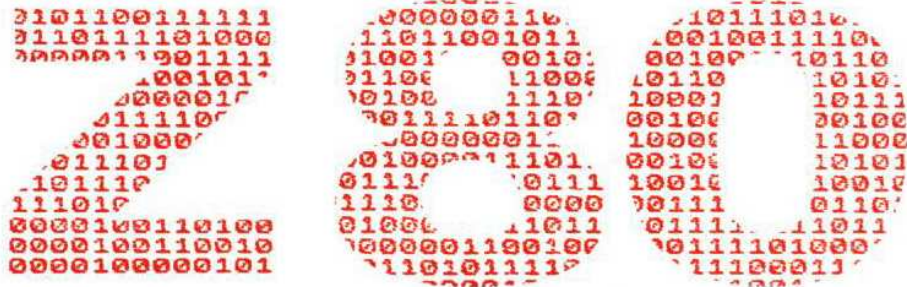
Envie seu pedido + cheque cruzado.
Prazo de entrega: 15 dias.
Despesas postais incluídas nos preços.

Atendemos somente por carta.
Solicite relação de programas em fita.

SIM, *desejo receber o* **TIG-LOADER**, *para tanto, estou anexando o cheque n.º* ______________ *no valor de Cr$* ______________

NOME:			ENDEREÇO:	
CEP:	CIDADE:	ESTADO:	PROFISSÃO:	ASSINATURA:
IDADE:	MARCA DO MICROCOMPUTADOR:			

CURSO DE ASSEMBLY
aula 5

Z80

Flavio Rossini

A INSTRUÇÃO LD PARA ENDEREÇAMENTO DIRETO E INDIRETO: CÓPIA DE REGISTROS NA MEMÓRIA E VICE-VERSA

Até agora, aprendemos como colocar números diretamente dentro dos registros internos do microprocessador e como copiar dados de um registro em outro. Vamos ver, agora, como copiar o conteúdo dos registros internos nos registros da memória e vice-versa. Veja como isto amplia nosso "Universo": até agora só podíamos colocar dados em 7 registros mas, com estas novas instruções, teremos à disposição toda a memória RAM! Naturalmente, cuidados devem ser tomados para não interferir com os demais programas ou a tela de TV; por enquanto este problema não nos atinge pois estamos utilizando uma região de memória "reservada" no fim da RAM.

Assim como em BASIC, escrever:

```
LET X=PEEK 30000
```

faz a variável X assumir um valor entre Ø e 255 que corresponde ao que está na memória 3ØØØØ; em linguagem de máquina existe uma instrução análoga mas que *só é válida para o registro A:*

LD A, (3ØØØØ) ou LD A, ('753Ø') cujo código é '3A' + 2 bytes para o endereço.

Note que os ***parênteses*** indicam a seguinte idéia: copie no registro A o ***conteúdo da memória 3ØØØØ***. De fato, a instrução LD A, 3ØØØØ (sem parênteses) NÃO é válida, porque não podemos colocar o nº 3ØØØØ dentro de 1 único registro, pois este número é maior do que 255. Este tipo de instrução é chamada de endereçamento DIRETO, pois podemos dizer ***diretamente*** na instrução, em que endereço da memória está o que queremos colocar no registro A.

Observe que, esta instrução equivale a ***3 bytes***: 1 para o código de operação (no caso '3A') e 2 para dizer qual o endereço cujo conteúdo devemos copiar no registro A (lembre-se da inversão).

Experimente então, colocar um número na memória. Por exemplo 162, na memória 3ØØØ7:

```
POKE 30007,162
```

e execute o seguinte programa:

MEM. 3ØØØØ	**LD A, (3ØØØ7)**	**'3A3775'**	**; (pois 3ØØØ7 = '7537)**
MEM. 3ØØØ3	**LD B,Ø**	**'Ø6ØØ'**	**; coloca Ø em B**
MEM. 3ØØØ5	**LD C,A**	**'4F'**	**; coloca A em C**
MEM. 3ØØØ6	**RET**	**'C9'**	**; volta ao BASIC**

Visualização do microprocessador.

(Obs.: colocamos o nº 162 na memória 3ØØØ7, por ser o 1º byte *livre,* após o programa e, como explicado anteriormente, é nessa região que colocaremos as variáveis que não cabem nos registros. Cuidado para não colocar números diretamente em memórias ocupadas pelo programa, pois você estará ALTERANDO as instruções).

3ØØØØ	ØØ111Ø1Ø	'3A'	LD A, (3ØØØ7)
3ØØØ1	ØØ11Ø111	'37'	
3ØØØ2	Ø111Ø1Ø1	'75'	
3ØØØ3	ØØØØØ11Ø	'Ø6'	LD B,Ø
3ØØØ4	ØØØØØØØØ	'ØØ'	
3ØØØ5	Ø1ØØ1111	'4F'	LD C,A
3ØØØ6	11ØØ1ØØ1	'C9'	RET
3ØØØ7	1Ø1ØØØ1Ø	'A2' = 162 (dado)	
.	....		
.	....		

Visualização da memória.

Esta subrotina copia o conteúdo da memória 3ØØØ7 no registro A e carrega o par BC com este número (note que fazemos B=Ø); portanto, na tela, deverá aparecer o número 162! O que aconteceria se retirássemos a instrução LD B,Ø? Note que seu valor continuaria sendo '75'; (pois 3ØØØØ = '753Ø') dessa forma, obteríamos:

117 x 256 + 162 = 3Ø114 ('75' = 117)!

Se, ao invés de PEEK-número, fizéssemos PEEK variável, por exemplo, LET X = PEEK Y (onde Y pode valer de Ø a 65535). Isto corresponde, em linguagem de máquina, uma instrução do tipo:

LD registro, (par de registros)

que significa "copie no registro da *esquerda* o *conteúdo* da memória *cujo endereço é indicado pelo par de registros à direita.*" No entanto, nem todas as combinações são possíveis e assim temos:

INSTRUÇÃO	CÓDIGO
LD A,(BC)	'ØA'
LD A,(DE)	'1A'
LD A,(HL)	'7E'
LD B,(HL)	'46'
LD C,(HL)	'4E'
LD D,(HL)	'56'
LD E,(HL)	'5E'
LD H,(HL)	'66'
LD L,(HL)	'6E'

A instrução LD registro — par de registros.

Este tipo de instrução é chamada endereçamento ***INDIRETO por pares de registro,*** pois o endereço daquilo que queremos colocar nos registros é dado indiretamente através dos pares BC, DE e HL para o registro A e somente do par HL para os demais registros (B, C, D, E, H, L).

Podemos perceber que o registro A parece ser "privilegiado" em relação aos demais; de fato, ele tem várias propriedades que os outros não têm e, inclusive, um nome especial: ACUMULADOR. Mais adiante, veremos com mais detalhes as propriedades especiais do ***ACUMULADOR.***

Coloque então, novamente, um número na memória 3ØØØ7:

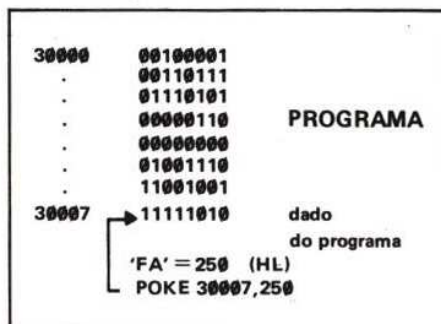
```
POKE 30007,250
```

e execute:

MEM. 3ØØØØ	LD HL,3ØØØ7	'213775'	; (3ØØØ7 = '7537') coloca o nº 3ØØØ7 no par HL
MEM. 3ØØØ3	LD B,Ø	'Ø6ØØ'	; coloca Ø em B
MEM. 3ØØØ5	LD C,(HL)	'4E'	; copia em C o conteúdo da memória apontada por HL (3ØØØ7)
MEM. 3ØØØ6	RET	'C9'	

Este programa, carrega o par HL com 3ØØØ7 e a seguir coloca em C, o conteúdo da memória que está endereçada por HL.

Note que precisamos fazer B = Ø pois o conteúdo da memória 3ØØØ7 tem apenas 8 bits e ocupará o registro C somente.

```
30000      00100001
  .        00110111
  .        01110101
  .        00000110        PROGRAMA
  .        00000000
  .        01001110
  .        11001001
30007   -> 11111010        dado
                           do programa
         'FA' = 250   (HL)
         POKE 30007,250
```

Assim como o PEEK tem sua "função inversa" (o POKE), as instruções que acabamos de ver também têm suas "funções inversas", a saber:

INSTRUÇÃO	CÓDIGO
LD (endereço),A	'32' (+2 bytes de endereço)
LD (BC),A	'Ø2'
LD (DE),A	'12'
LD (HL),A	'77'
LD (HL),B	'7Ø'
LD (HL),C	'71'
LD (HL),D	'72'
LD (HL),E	'73'
LD (HL),H	'74'
LD (HL),L	'75'

Note que a primeira instrução desta lista corresponde a 3 bytes, enquanto que as demais apenas um! Perceba novamente como o ***ACUMULADOR*** é privilegiado: ele é o único que permite endereçamento DIRETO colocando o número desejado da memória e, ao contrário dos demais, permite endereçamento INDIRETO não só através do par HL, mas também BC e DE!

Vamos então fazer mais um programa para exemplificar o que acabamos de aprender: no início do curso, fizemos referência às variáveis do ***programa interpretador*** que estão no início da memória RAM (entre 16384 e 165Ø9) e exemplificamos, dizendo que a variável ***RAMTOP*** tinha 2 bytes e estava nos endereços 16388 e 16389; vamos agora utilizar ***outra variável de apenas 1 byte*** e que está no endereço 16442. Ela indica a posição da linha PRINT (em BASIC), com um número de 1 a 24; como sabemos, a tela tem 22 linhas acessíveis aos programas BASIC mais 2 linhas de edição. É considerada a linha 1 a linha "mais baixa" da tela; experimente executar (RUN 4ØØØ):

```
4000 SLOW
4005 FOR I=0 TO 21
4010 FOR J=0 TO 3
4015 PRINT TAB (8*J);PEEK 16442
4020 NEXT J
4025 NEXT I
```

Programa 1

Vamos então "simular" este PEEK com uma subrotina ou linguagem de máquina:

3ØØØØ	**LD A,(16442)**	**'3A3A4Ø'**	**; (16442 = 4Ø3A)**
3ØØØ3	**LD B,Ø**	**'Ø6ØØ'**	
3ØØØ5	**LD C,A**	**'4F'**	
3ØØØ6	**RET**	**'C9'**	

a qual "carrega o par de registros BC com o conteúdo da memória 16442. Modifique então a linha 4Ø15 do programa BASIC:

```
4015 PRINT TAB (8*J);USR 30000
```

Execute o programa e você deverá obter o mesmo resultado anterior! Na realidade, esta subrotina faz exatamente o que a função PEEK faz: "coloca no ***ACUMULADOR*** o conteúdo da memória 16442 e, para obter o resultado na tela, copia o conteúdo do acumulador em C, fazendo B = Ø."

Na memória 16441 está outra variável de 1 byte que indica o número da ***coluna*** para PRINT. Experimente fazer um programa similar ao anterior, mas que use uma subrotina em linguagem de máquina, para fazer o PEEK 16441.

Feito isto, tente agora simular a instrução POKE em linguagem de máquina. Como sugestão, procure fazer um programa em BASIC cheio de PRINTs com pausas entre os mesmos. Interrompa o programa usando BREAK e a seguir altere o valor das memórias 16427 e 16428. Nestas memórias fica guardada a linha para a qual o programa deve ir, após ser pressionado o CONT.

Faça as alterações usando POKEs diretamente e com um programa em linguagem de máquina (lembre-se da inversão para os números maiores que 255). Verifique então se após o CONT o programa vai para a linha que você especificou (por que esta variável deve ter 2 bytes?).

Iremos agora completar a "lista" de instruções LD usando endereçamento ***direto*** para pares de registros. O que você acha que significa a instrução:

LD HL, (16388)?

Como é possível copiar o conteúdo da memória 16388 que tem 1 byte apenas, no par de registros HL que tem 2 bytes? De fato, esta instrução funciona da seguinte maneira: copia no registro L (o menos significativo) o conteúdo da memória 16388 e no registro H (o mais significati- →

vo) o conteúdo da memória 16389. Isto em BASIC poderia ser associado a:

```
LET X=PEEK 16388+256*PEEK 16389
```

(você está lembrando qual a variável contida nessas memórias?)

Note bem a diferença entre as seguintes instruções:

LD HL,16388
LD HL,(16388)

A segunda foi a que acabamos de descrever; a primeira foi descrita anteriormente e significa: carregue o par HL com o *número* 16388, ou seja, H = 'Ø' e L = 'Ø4'.

Naturalmente existe também a instrução inversa:

LD (16388),HL

que copia o registro *L* na memória 16388 e o *H* na memória 16389 e que em *BASIC* equivaleria a:

```
POKE 16388,X-INT (X/256)*256
POKE 16389,INT (X/256)
```

você saberia explicar por quê?

Assim temos as seguintes instruções:

INSTRUÇÃO	CÓDIGO		
LD BC, (endereço)	'ED4B'	+	(2 bytes para o endereço)
LD BE, (endereço)	'ED5B'	+	(2 bytes para o endereço)
LD HL, (endereço)	'2A'	+	(2 bytes para o endereço)
LD (endereço), BC	'ED43'	+	(2 bytes para o endereço)
LD (endereço), DE	'ED53'	+	(2 bytes para o endereço)
LD (endereço), HL	'22'	+	(2 bytes para o endereço)

Note que aqui aparecem algumas instruções cujo código de operação tem 2 bytes. Para todas elas o primeiro byte corresponde a *'ED'*. Isto porque se todos os códigos de operação tivessem apenas 1 byte, seria possível ter apenas 256 códigos diferentes! Portanto, para ampliar o número de instruções, algumas têm o mesmo código que as outras precedidas por *'ED'*, outras ainda, precedidas por *'CB'*. Por exemplo, o código *'4B'*, sózinho, corresponde à instrução (*LD C,E*) e, se for precedido por *'ED'*, ou seja, *'ED4B'*, corresponde a *LD BC*, (endereço).

Para finalizar, vamos apresentar a instrução:

LD (HL),número (código '36'),

que permite colocar um número de *1 byte diretamente* na memória cujo endereço é dado pelo par *HL*. (*Apenas o par HL permite esta instrução*!)

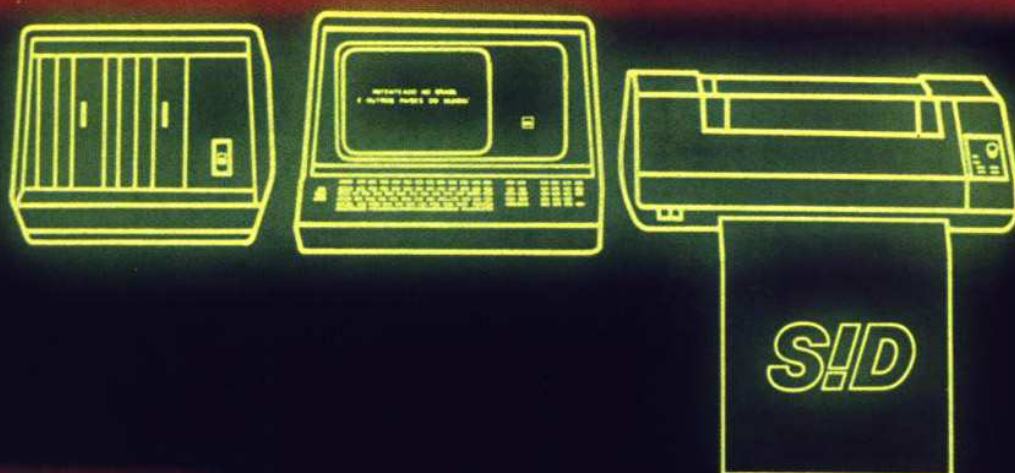
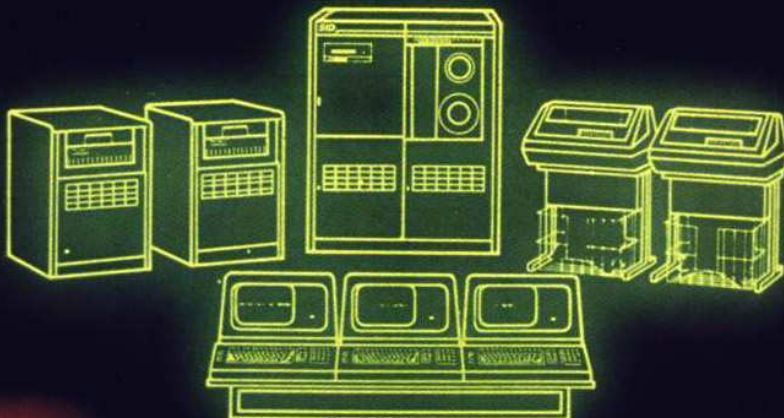
Note que, não existe instrução para copiar DIRETAMENTE um registro de memória para outro, assim como fazemos nos registros internos! E assim para copiar, por exemplo, o conteúdo da memória 3Ø5ØØ na memória 3Ø51Ø é necessário que o dado seja antes COPIADO num registro interno do microprocessador, de preferência o ACUMULADOR. Bastaria então fazer:

```
LD A, (30500)   '3A2477'   ; (30500 = '7724')
LD (30510),A    '322E77'   ; (30510 = '772E')
RET             'C9'
```

Teste o programa, fazendo por exemplo POKE 3Ø5ØØ,31 e, após executar o programa, fazendo PRINT PEEK 3Ø51Ø; onde você deverá obter o número 31.

Anteriormente dissemos que o único registro que permitia um *LD direto* de um endereço da memória era o ACUMULADOR:*LD A,* (endereço).

Podemos, com um pequeno truque, fazer o mesmo para os registros *C, E* ou *L;* observe um exemplo para o registro *E:* vamos simular a instrução (*LD E,*-endereço) que de fato NÃO existe:

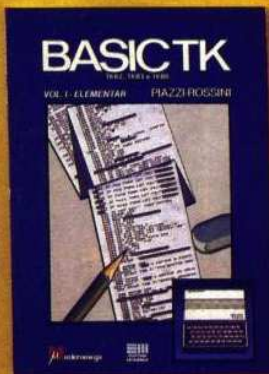
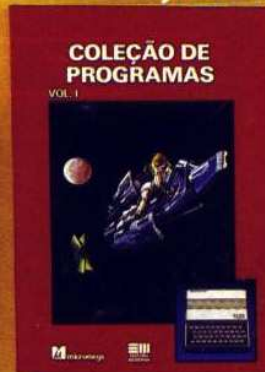
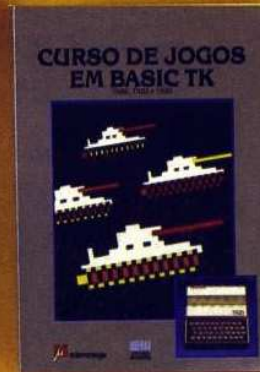
O MICRO E O MINICOMPUTADOR DO ADMIRÁVEL MUNDO NOVO SID.

MICROCOMPUTADOR SID 3000. Um verdadeiro pequeno grande profissional. Porque o SID 3000 foi concebido para oferecer uma configuração compacta e altamente modular, além de uma extensa gama de sistemas aplicativos que permitem a máxima eficiência e rapidez na prestação de serviços administrativos. Seja qual for o porte ou atividade da empresa. Contabilidade, Folha de Pagamento, Faturamento, Contas a Pagar/Receber, Controle de Estoque, Ativo Fixo, Livros Fiscais, são alguns dos aplicativos disponíveis, como também sistemas específicos para Hotéis, Escolas, Distribuidoras de Bebidas, Cooperativas, Hospitais, Empresas de Transporte Urbano, Supermercados, Transportadoras, Imobiliárias, Administração Municipal, Controle de Carnês, Leasing, Renda Fixa, Open Market, Escritórios de Contabilidade. E, caso estes sistemas não se apliquem às necessidades do usuário, a SID coloca à disposição sua estrutura técnica para o desenvolvimento de novos sistemas.
O importante é encontrar a solução exata para racionalizar o tempo/serviço da empresa. Portanto, quando se trata de analisar custo e performance, a SID pode responder às exigências do mercado com a melhor alternativa em microcomputadores: SID 3000.

MINICOMPUTADOR SID 5000. Quando as necessidades de uma empresa estão voltadas para um mini, com certeza o SID 5000 é a solução definitiva. Destinado a aplicações de maior porte, apresenta uma superior capacidade de armazenamento de dados, possibilitando a conexão de um maior número de terminais, de acordo com o aumento do volume de informações.
A série 5000 compõe-se de diferentes modelos, concebidos de forma modular e compatíveis entre si, permitindo que o sistema tenha um crescimento gradativo em função de novas necessidades. Sua avançada concepção apresenta características antes só encontradas em equipamentos de grande porte, como processamento multitransacional e memória virtual.
SID 5000. O máximo de tecnologia em minicomputadores.

O Admirável Mundo Novo da Informação

SHARP

FILIAIS: BELÉM (091) 226.9877 • BELO HORIZONTE (031) 225.9400 - 225.3711 • BRASÍLIA (061) 245.3388 • CAMPINAS (0192) 51.3283 • CURITIBA (041) 224.7522 • FORTALEZA (085) 231.5711 • GOIÂNIA (062) 225.0681 • PORTO ALEGRE (0512) 25.9488 - 33.7161 • RECIFE (081) 221.4828 - 231.3362 - 231.1403 • RIO DE JANEIRO (021) 591.2242 - 591.0096 • SALVADOR (071) 242.8680 - 243.6166 • SÃO PAULO - FILIAL BANCOS E GOVERNO (011) 255.1205 - 256.3033 • FILIAL COMÉRCIO E INDÚSTRIA (011) 255.1027 - 256.3033

Quatro maneiras para você utilizar melhor o seu micro.

(Para usuários de TK 82 c, TK 83, TK 85, NZ 8000, CP 200, ZX 81 e TIMEX 1000).

Curso didático de linguagem Basic para iniciantes, com muitos exemplos e exercícios.

Dezenas de programas para instrução e lazer em dois volumes.

Divirta-se aprendendo truques de programação.

Ensina linguagem de máquina para você tirar o máximo proveito do seu micro.

Desperte as habilidades secretas de seu micro!

Não deixe de ler estes livros.

micromega
Publicações e Material Didático Ltda.
fone: (011) 256-8348

Sim quero receber os livros assinalados abaixo: (faça um "X").

- ☐ Linguagem de máquina para o TK. Cr$ 6.500,00
- ☐ Curso de jogos em Basic TK. Cr$ 3.200,00 (Jan. 84)
- ☐ Coleção de programas Vol. I. Cr$ 3.200,00
- ☐ Coleção de programas Vol. II. Cr$ 3.400,00
- ☐ Basic TK. Cr$ 4.500,00 (Jan. 84)

Valor total Cr$ 20.800,00

Envie seu cheque nominal e cruzado, ou vale postal para Micromega P.M.D. Ltda. Caixa Postal 54121 — CEP 01296 — São Paulo — SP

Nome: ______
Endereço: ______
CEP: ______ Cidade: ______
Estado: ______ Data: ______
Assinatura: ______

Válido até 31/12/83

BASIC E LINGUAGEM DE MAQUINA

1984... você está preparado?

NÍVEL Ø	NÍVEL 1	NÍVEL 2A	NÍVEL 2B	NÍVEL 3	NÍVEL 4
CRIANÇAS	ADULTOS E ADOLESC.	ADULTOS E ADOLESC.	ADOLESC. JOGOS	ADULTOS E ADOLESC.	ADULTOS E ADOLESC.
10 h	20 h	20 h	20 h	40 h	40 h
INICIAÇÃO	INICIAÇÃO	APROF. EM APLICAÇÕES ADM.	APLICAÇÕES EM JOGOS	LINGUAGEM DE MÁQUINA ASSEMBLY Z80	APROF. DE LINGUAGEM DE MÁQUINA

Núcleo de orientação de estudos

CORPO DOCENTE:
PROF. FÁBIO RENDELUCCI
PROF. FLAVIO ROSSINI
PROF. PIERLUIGI PIAZZI

Av. Brig. Faria Lima, 1.451 - 3º - Cj. 31
Tel.: 813-4555 - CEP 01451 - São Paulo

MEM. 30000	LD DE, (30010)	'ED5B3A75'	; copia em E o conteúdo da memória 30010 e em D o da memória 30011
MEM. 30004	LD D,0	'1600'	; "zera" o registro D
MEM. 30006	LD B,0	'0600'	; "zera" o registro B
MEM. 30008	LD C,E	'4B'	; copia o registro E em C
MEM. 30009	RET	'C9'	

Faça POKE 30010,100 e você deverá obter 100 na tela ao executar o programa. (*Nota:* lembre-se sempre que estamos usando para colocar as *variáveis* destes programas em linguagens de máquina, a região de memória *após* o fim dos mesmos; por exemplo, o último programa tinha 10 bytes e portanto, começava na memória 30000 e terminava na memória 30009; assim colocamos nossa variável na memória 30010. MUITO CUIDADO DEVE SER TOMADO PARA NÃO ERRAR NESTE PROCEDIMENTO POIS, CASO CONTRÁRIO, VOCÊ PODERÁ MODIFICAR O PRÓPRIO PROGRAMA OU FAZER UM PROGRAMA QUE USA SUAS PRÓPRIAS INSTRUÇÕES COMO VARIÁVEIS (OU DADOS) E NOVAMENTE IRÃO APARECER AS COISAS ESTRANHAS! Por exemplo, se no programa anterior você fizesse POKE 30009, 100, você estaria substituindo a instrução RET código 'C9'(201) pelo número 100; o computador, não achando a instrução RET, não poderia mais voltar ao programa em BASIC!)

EXERCÍCIOS

1) Faça um programa em linguagem de máquina que copie o conteúdo da memória 30015 na memória 30016. Primeiramente, coloque o número 30015 no par *HL* e o número 30016 no par *DE*. Feito isto, copie o conteúdo da memória indicada por *HL* no ACUMULADOR e, a seguir, copie o conteúdo do acumulador na memória indicada por *DE*. A seguir, copie o ACUMULADOR no registro *C* e "zere" o registro *B*. Assim, ao executar o programa você irá obter na tela, o número que estava na memória 30015.

Obviamente, antes de executar coloque um número na memória 30015, por exemplo: POKE 30015,99 e, uma vez executado o programa, verifique se ele funcionou fazendo: PRINT PEEK 30016 e você deverá obter 99!

(Observação: Existem meios mais simples do que foi sugerido para fazer este programa; no entanto achamos esta maneira a mais didática no atual estágio do curso).

2) Responda às seguintes perguntas:

a) Por que nas instruções de *LD* nunca aparecem registros simples entre parênteses, apenas pares de registros?

b) Se o par *HL* contém o valor 16434, qual a diferença entre as instruções *LD B, (HL), LD BC,* (16434) e *LD BC,* 16434?

c) Tente fazer um programa em linguagem de máquina que copie no par de registros *HL,* o valor da memória 16442 (note que *H* deverá, portanto, ser zero!), usando, para lidar com a memória, apenas instruções do tipo *LD* registro, (*HL*).

d) Suponha que o acumulador contenha o número 24 e que o par *HL* também contenha 24 (H = '00' e L = '18'); procure analisar o efeito das seguintes instruções:

```
LD A, 24      LD A, (24)
LD H, 24      LD A, (HL)
LD L, 24      LD (24), A
LD HL, 24     LD (HL), A
LD L, A       LD HL, (24)
LD A, H       LD (24), HL
              LD (HL), 24
```

e) *se existisse,* qual seria o efeito da instrução *LD BC, (HL)*?

PROTEÇÃO JURÍDICA DO PROGRAMA DE UM COMPUTADOR

2ª PARTE

Eduardo Vieira Manso

Sempre que não existe uma norma jurídica específica, para aplicação a um caso concreto (quer dizer, um caso real, que verdadeiramente está acontecendo e não um caso meramente hipotético, em tese), o Direito autoriza o aplicador das leis (os Juízes, em nosso sistema jurídico) a recorrer à analogia, ou aos costumes ou, então, aos chamados princípios gerais de direito. Princípios gerais de direito todos sabem o que são, mas ninguém consegue definir com exatidão, embora qualquer pessoa os reconheça tão logo esteja em frente de um deles, graças à consciência de justiça que todo homem naturalmente possui.

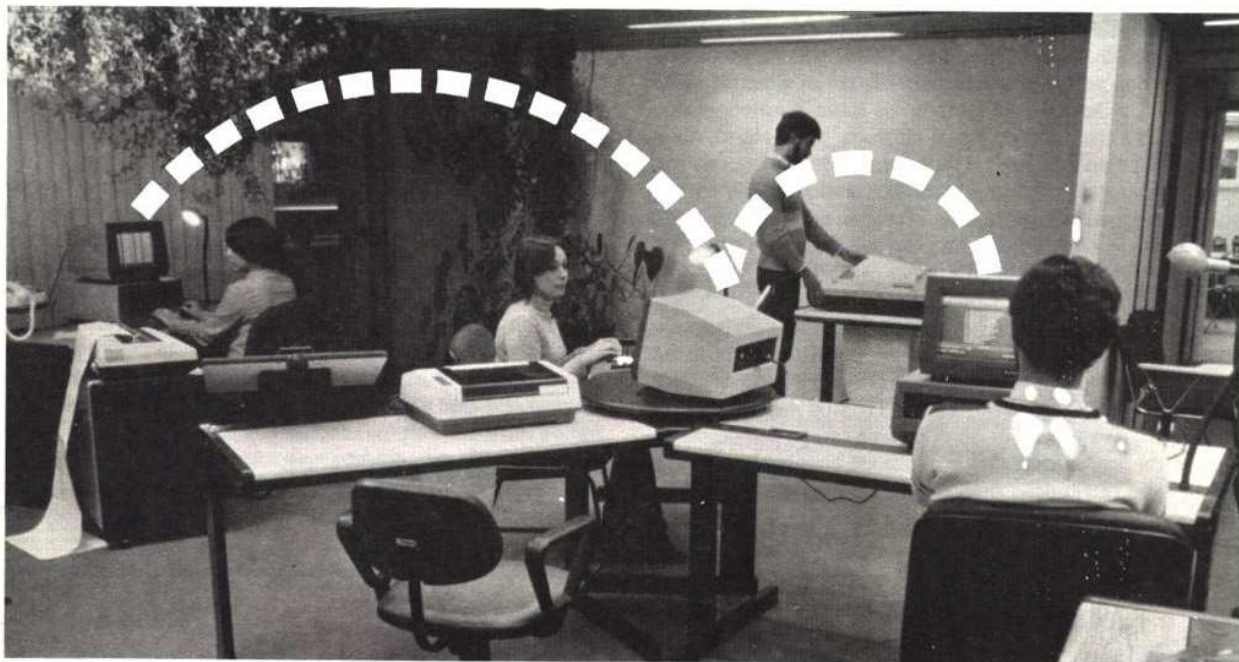
Desde logo, como até parece óbvio, não se haverá de encontrar possibilidade de recurso aos costumes jurídicos, atinentes aos casos que envolvem pirataria de programas de computadores eletrônicos. Nem aqui, nem em qualquer outro lugar do mundo, praticamente. Não houve, ainda, tempo suficiente para que se formassem costumes sobre tais questões, nem há bastante vivência de tais problemas, nos limites da jurisprudência, para que pudessem servir de fonte subsidiária do direito sobre programas.

No entanto, com base na analogia, combinada com os princípios gerais de direito, será possível lograr-se uma proteção razoável, para os casos de desrespeito desses interesses que hão de estar ao abrigo da Justiça, para que se impeçam enriquecimentos ilícitos e injustos; para que se combata a concorrência parasitária, a qual constitui o injusto e ilícito aproveitamento do trabalho alheio, para fins econômicos.

Ora, o programa de computador eletrônico é, sem nenhuma dúvida, uma obra intelectual, porquanto é o resultado material e formal da aplicação do intelecto no mundo das idéias. Prova suficiente dessa afirmação está no fato de que os Estados Unidos da América os protegem com as normas próprias do seu ***copyright,*** tanto quanto as Filipinas os inscrevem como obras protegidas pelo seu Direito de Propriedade Intelectual.

Na França, já houve um julgamento, proferido pela Quarta Câmara da Corte de Apelação de Paris, em 02.11.82, considerando que ***a elaboração de um programa de aplicação de computador eletrônico é uma obra de espírito original em sua composição e em sua expressão, indo além de uma simples lógica automática e obrigatória; que não se trata de um mecanismo intelectual necessário*** (no sentido de que não poderia ter sido elaborado de outra maneira, nem de outra forma); ***que, com efeito, os analistas-programadores devem escolher, como os tradutores de obras, entre diversos modos de apresentação e de expressão e que sua escolha, assim, traz a marca de sua personalidade.*** Naquele caso, tratava-se de um chefe de contabilidade, não contratado como programador, que elaborou diversos programas logo que sua empregadora adquiriu um computador eletrônico e que se insurgiu contra a vontade da empresa de duplicar aqueles programas. Por isso, o contador levou seus programas para casa, em razão de que foi despedido, sob alegação de ter cometido falta grave. Na primeira instância, a Décima Quinta Câmara do Tribunal de Comércio de Paris entendeu não ter havido tal falta grave, condenando a empregadora no pagamento das indenizações trabalhistas. Esse julgamento é que foi confirmado pela Corte de Apelação parisiense, quando houve aplicação de normas do direito autoral: na França, se a obra intelectual é criada por um empregado contratado para criar obras intelectuais, os direitos autorais pertencem originalmente ao empregado, cabendo ao empregador direitos de uso, salvo um contrato escrito contendo cessão de direitos. (No Brasil, a situação é bastante diferente: o direito autoral sobre obra criada por empregado contratado para criá-la caberá ao empregador e ao empregado, em cotitularidade, salvo se houver alguma disposição em contrário, no contrato de emprego, ou em outro documento). O Tribunal de Paris, que julgou a apelação daquela empregadora, deu, aos programas elaborados pelo ex-empregado, o mesmo estatuto jurídico que a lei defere aos livros, nada obstante também na França a lei de 1957 sobre Direito Autoral nada fale, expressamente, sobre programas de computadores.

REDE:

ANUNCIAMOS O FIM DO MICROCOMPUTADOR ISOLADO.

Finalmente, o processamento de dados da sua empresa deixou de ficar limitado ao trabalho de estações isoladas.

Os novos microcomputadores Dismac podem ser ligados entre si e a computadores de maior porte, compartilhando arquivos, impressoras e outros periféricos comuns.

As mais variadas consultas podem ser feitas ao mesmo tempo, com rapidez e segurança.

E sua empresa pode entrar nesse sistema começando pela configuração mono ou multiusuário, pois o sistema Rede pode ir crescendo à medida das suas necessidades, expandindo-se até 8 terminais inteligentes trabalhando com 2 microprocessadores de alto desempenho e até 576 Kbyte em CPU's independentes.

Uma série de aplicativos, especialmente desenvolvidos para o processamento distribuído, dão à sua empresa a agilidade operacional que ela precisa.

Entre em Rede. O fim do microcomputador isolado. O começo das soluções integradas.

Conheça também a linha de computadores pessoais Dismac 8100, compatível com Apple II Plus, nas lojas especializadas e revendedores em todo o Brasil.

nome ______________________

empresa ______________________

cargo ______________________

endereço ______________________

telefone ______________________

cidade/estado ______________________

☐ material explicativo
☐ demonstração em seu show-room

Manaus R. Içá, n.º 100 - Distrito Industrial - F. (092) 237-2474 • **São Paulo** - Av. Marquês de São Vicente, n.º 600 - F. (011) 826-7111
Representantes em todo o território nacional.

Assim, em face desses precedentes legais, dos Estados Unidos e das Filipinas, e jurisprudencial, da França, entendo que também os brasileiros poderão obter alguma proteção contra a violação de seus interesses, nos casos de pirataria principalmente, com base nas regras do nosso Direito Autoral, aplicadas analogicamente, e com fundamento no princípio de que a ninguém é dado locupletar-se com o esforço alheio, sem que ao menos retribua parte desse locupletamento.

Portanto, enquanto não houver um conjunto de normas jurídicas especialmente editadas para a proteção dos programas de computadores eletrônicos, é possível recorrer-se à Justiça, com base nas normas relativas à proteção das obras intelectuais, que são objeto dos direitos autorais, em face da perfeita analogia até mesmo com as Disposições Tipos da OMPI. Por sinal, a doutrina italiana, quase sem dissidência, considera o programa de computador uma obra análoga àquela protegida pelo direito conexo ao direito autoral, nos termos do que dispõe o art. 99, da Lei nº 633, de 22 de abril de 1941, a despeito de também na Itália se entender que uma tal proteção é insuficiente e mal ajustada. Com efeito, diz o artigo 99 da lei italiana de 1941: "Aos autores de projetos de trabalho de engenharia, ou de outros trabalhos análogos que constituam soluções originais de problemas técnicos, compete, além do direito exclusivo de reprodução dos planos e desenhos dos mesmos projetos, o direito a uma equitativa retribuição a cargo daqueles que executem o projeto técnico com finalidade de lucro, sem seu consentimento." Como se vê, tal proteção fundada naquele tipo de direito conexo, análogo ao direito autoral seria mesmo insuficiente, porque, contra a reprodução não autorizada, ou o uso não autorizado, com fim de lucro (o que restringe a proteção, por certo), dá-se ao autor do programa (ou ao seu cessionário), como se dá aos autores dos projetos de engenharia, apenas um direito a uma retribuição pelo uso não autorizado, ou cópia pirata. Isso, entretanto, é o mesmo que instituir uma espécie de pirataria oficializada pela remuneração posterior: poder-se-ia usar o programa, pagando depois desse uso e, mesmo assim, se houver alguma reclamação contra ele. Porém, como quem não tem cão deve mesmo caçar com gato, as regras atuais do nosso Direito Autoral podem e devem ser aplicadas, analogicamente, nos casos de reclamação judicial contra toda modalidade de uso indevido de programas alheios. Com isso, estar-se-á cumprindo, ainda que parcialmente, um dos mais indisputáveis objetivos dos governos da sociedade, que é a distribuição da Justiça.

No entanto, para que se alcance essa proteção subsidiária, não é necessário que se tome nenhuma providência de ordem administrativa, tal como efetuar registros públicos, por exemplo, embora essas providências sejam úteis, especialmente para a prova da autoria dos programas. Por isso, o registro de programas, instituído pela SEI nos termos de seu Ato Normativo nº 22, de 02.12.

82, sem embargo de não ter essa finalidade, será extremamente valioso para ser feita a comprovação da autoria, o que é fundamental em qualquer espécie de disputa judicial.

Em conclusão: acredito que, mesmo em se tratando de aplicação analógica das regras do Direito Autoral, que estão consubstanciadas na Lei nº 5.988, de 14 de dezembro de 1973, particularmente nos arts. 29, 30, 31, 32 (exceto, talvez, seu parágrafo único) e art. 40 (o art. 36 também haverá de ter alguma aplicação, com algum temperamento, em face das peculiaridades da obra que é o programa e de sua gênese, que quase sempre ocorre no seio de uma pessoa jurídica), é imperioso que os programadores ou os cessionários dos seus direitos sobre os programas comecem a pleitear em juízo a proteção de seus interesses.

E acredito que, principalmente, podem ser pleiteadas as seguintes prestações jurisdicionais:

a) a salvaguarda do direito de reproduzir o programa mediante cópias idênticas, sejam estas feitas a partir de programas vertidos para a linguagem de máquina, sejam em qualquer outra linguagem;

b) o direito de autorizar a utilização dos seus programas, ou, ao menos, o de receber alguma remuneração, além daquela prevista nos contratos com base nos quais aqueles programas tenham sido elaborados, quando a utilização também se der além dos limites contratuais. Assim, por exemplo, um empregado que tenha como função a elaboração de programas, para uma empresa que não comercializa programas, deverá ter direito de, no mínimo, participar dos resultados econômicos que essa empresa tiver de qualquer eventual comercialização de programas feitos em razão do contrato de emprego. No entanto, o empregado contratado para elaborar programas para uma empresa que tenha como objetivo a sua comercialização, não haverá de ter direito de participação nos resultados de tal comercialização, sendo de supor-se que tais programadores terão ajustado salários, honorários, em que tal hipótese já tenha sido levada em conta;

c) o direito de apreender cópias piratas de seus programas, quando suficientemente comprovada a autoria delas e o servilismo das cópias.

Somente se terá pefeita noção do alcance dos direitos de que os programadores poderão dispor, hoje, com base na analogia e nos princípios gerais de direito, a partir do momento em que os casos forem levados às barras da Justiça, de preferência através do patrocínio de advogados especializados. Mesmo a proteção por via de contratos muito bem elaborados não será segura, enquanto sobre eles os nossos juízes não se pronunciarem em processos regularmente levados a efeito. Em tal estado de coisas, apenas uma sentença judicial, irrecorrível, é que dará certeza de todo o direito de que alguém desfruta, em face de violações disso que já podemos dizer serem seus direitos sobre o logiciário.

EDUARDO VIEIRA MANSO é especialista em Direitos Intelectuais, principalmente em Direito Autoral, com larga experiência como consultor jurídico e advogado das principais empresas de comunicação do Brasil. Já publicou diversos artigos sobre sua especialidade em jornais e revistas nacionais e estrangeiras, além de duas monografias pioneiras no Brasil: "A OBRA DO AUTOR ASSALARIADO" (edição da Abril S/A, São Paulo, 1975) e "DIREITO AUTORAL: Exceções Impostas aos Direitos Autorais -- Derrogações e Limitações" (edição José Bushatsky, São Paulo, 1980). Tem ministrado aulas como professor convidado das Faculdades de Direito da USP, da PUC/SP e do Mackenzie, e cursos monográficos que organizou para o IDORT e para a ASSOCIAÇÃO DOS ADVOGADOS DE SÃO PAULO, entre outros, além de palestras no Brasil e no Exterior.

CONTABILIDADE I

A fita deste mês prima por uma qualidade de real valor: "a marca do produto nacional". É uma fita fabricada por uma empresa brasileira (A Micron Eletrônica) sediada em Campinas, São Paulo, podendo ser adquirida em diversas lojas da área, no valor de 4 mil cruzeiros.

Utilizando 6k de memória, a fita foi feita com o objetivo de atingir os interesses das pequenas empresas e escritórios de contabilidade que prestam serviços a elas.

O que nos levou à análise desta fita — além da indiscutível qualidade e praticidade do programa — foi o fato de que ela é fruto e prova do real esforço que tem sido feito pelos fabricantes de software e hardware, no sentido de implementar e desenvolver (a ponto de comparar-se, na área de informática, aos países desenvolvidos) uma tecnologia própria adequada à realidade do Brasil. É um esforço digno e deve ser louvado, principalmente quando pensamos na importância da chamada "reserva de mercado" pois, pressões do exterior é que não faltam pela conquista deste mercado.

A fita chegou às nossas mãos acompanhada da listagem, onde constam o programa com as instruções e as várias etapas do cálculo de uma folha de pagamento.

Testamos a fita em vários gravadores e tiramos a prova de sua qualidade, ao observarmos, a gravação sem problemas e livre de interferências. Ao colocarmos a fita no TK ficamos preocupados em comprovar a precisão de cálculo do programa, já que a legislação da área possibilita um pouco de imprecisão no que tange as limitações de um micro. Mas logo chegamos à conclusão de que a fita, por seu teor prático, seria de excelente utilidade a nossos leitores, principalmente aqueles que se interessam pela área administrativa, além do que possibilita a qualquer leigo (que trabalha na área, mas que não tem prática no cálculo da folha de pagamento e no recolhimento de tributos), gravar o programa e efetuar o cálculo do IAPAS, naqueles momentos críticos do fim do mês.

A listagem-exemplo enviada pela Micron nos mostra a folha de pagamento, por funcionário, onde constam o valor bruto do rendimento, descontos, bonificações (horas-extras, salário-família e prêmios) e o total líquido a receber. Logo a seguir surge o total da folha de pagamento do mês de onde será calculada, posteriormente, a guia de recolhimento do IAPAS.

Continuando, a listagem (bastante extensa e detalhada) nos mostra o cálculo da guia do IAPAS. O cálculo desta guia é efetuado por *faixa salarial* e o programa vai dando o valor de cada código constante da guia, os sub-totais e, no final, o valor a recolher de cada *faixa.*

O final da listagem mostra o preenchimento da "guia única" onde se observam os totais gerais de cada código, ou seja, o total do INPS a recolher naquele mês; a seguir surge o valor, também a recolher no mês, do FGTS.

Para se fazer todos os cálculos, o programa se distribui em três etapas, fornecendo três códigos para chamar cada uma delas. O primeiro — "código 1", possibilita o início dos cálculos; o segundo — "código 2" —, possibilita a gravação do programa e este apresenta: a folha de pagamento, as guias de IAPAS e FGTS (sem os contra-cheques). O "código 3", nos dá o resumo da folha de pagamento, fornecendo o total a recolher tanto do INPS como do FGTS (como já dissemos acima).

Como já comentamos no início, ***Contabilidade I — Folha de Pagamento*** é uma fita que mereceu nossa atenção na sua escolha para publicação, tanto por seu caráter de fabricação e idealização nacionais, como também pelo caráter profissional que possibilita sua utilização por qualquer pessoa, desde que tenha, às mãos, uma guia de recolhimento para que saiba os códigos existentes na guia e esta pessoa possa fornecê-los ao computador, e este possa calcular os seus respectivos valores.

Cabe ressalvar que, de nossa parte, é uma inovação a publicação desta fita nesta coluna de **Microhobby** pois, desta maneira, estamos dedicando a seção ***"Fita do Mês"*** também para aquelas fitas que se voltam para a utilização do micro por profissionais nas diversas áreas do conhecimento humano.